CHAPTER 9

Lossless Image Compression

K. P. SUBBALAKSHMI

OVERVIEW

Lossless image compression deals with the problem of representing an image with as few bits as possible in such a way that the original image can be reconstructed from this representation *without* any error or distortion. Many applications such as medical imaging, preservation of artwork, image archiving, remote sensing, and image analysis call for the use of lossless compression, since these applications cannot tolerate any distortion in the reconstructed images. With the growth in the demand for these applications, there has been a lot of interest lately in lossless image compression schemes. In this chapter, we discuss some of the basic techniques used in lossless image compression.

9.1 INTRODUCTION

An image is often viewed as a two-dimensional array of intensity values, digitized to some number of bits. In most applications 8 bits are used to represent these values, hence the intensity values of a gray-scale image can vary from 0 to 255, whereas in radiology, the images are typically quantized to 12 bits.

It is possible to apply some of the entropy coding schemes developed in the earlier chapters for lossless image compression directly to an image; however, this can be expected to result in only moderate compression. This is because sources like images exhibit significant correlation between adjacent pixels, as can be seen from a visual inspection of a natural image like "Lenna" shown in Fig. 9.1. By directly entropy coding the image, we do not use the substantial structure that exists in the source. Using this inherent structure in the image will let us compress it to a larger extent. Indeed most state-of-the-art lossless image compression techniques exploit this correlation.

FIGURE 9.1
Grayscale image "Lenna."

Existing lossless image compression algorithms can be classified broadly into two kinds: those based on spatial prediction and those that are transform based. Both of these approaches can be looked upon as using the correlation between the consecutive pixels in an image to compress it. In prediction-based methods, the current pixel is predicted from a finite context of past pixels and the prediction error is encoded. These techniques are usually applied to the time domain representation of the signal although they could potentially be used in the frequency domain. Transform-based algorithms, on the other hand, are often used to produce a *hierarchical* representation of an image and work in the frequency domain. A hierarchical or *multiresolution* algorithm is based on creating a coarse representation and then adding refinement layers to it. This coarse representation can be used to give a "preview" of the image to the user. Hence such algorithms are very useful in large image database browsing applications, where the coarse resolution version can be used to select the desired image from a huge selection. Hence, transform-based lossless compression has been more popular in applications where a spectrum of compression accuracies (from lossless to lossy) is desired rather than in applications where only lossless compression is preferred. With the success of the latest wavelet-based lossy compression standard, JPEG 2000, transform-based lossless compression methods could become more useful in the future.

This chapter is divided into two parts; the first part introduces some of the basic ideas used in lossless compression and the second delves more deeply into the specific details of algorithms that have been designed for lossless image compression over the years.

9.2 PRELIMINARIES

Entropy coding is more efficient for sources that are distributed less evenly. In other words, when the *probability mass function* (pmf) of a source is skewed or when some of the source outputs occur significantly more often than the others, the source can be more efficiently compressed using entropy codes. Most natural images like Lenna, in Fig. 9.1, do not have such skewed pmf's to begin with (see Fig. 9.2); however, there are ways of extracting sequences with such pmf's from the original images. In sources that are correlated, one of these ways is to predict the current sample value from the past sample values and code the error in prediction. Another way is to predict the pmf of the current pixel based on some of its immediate context. Prediction-based algorithms are discussed in Section 9.2.1.

As mentioned in the Introduction, hierarchical coding of images can be useful in database browsing applications. The basic idea behind this method is to generate several descriptions

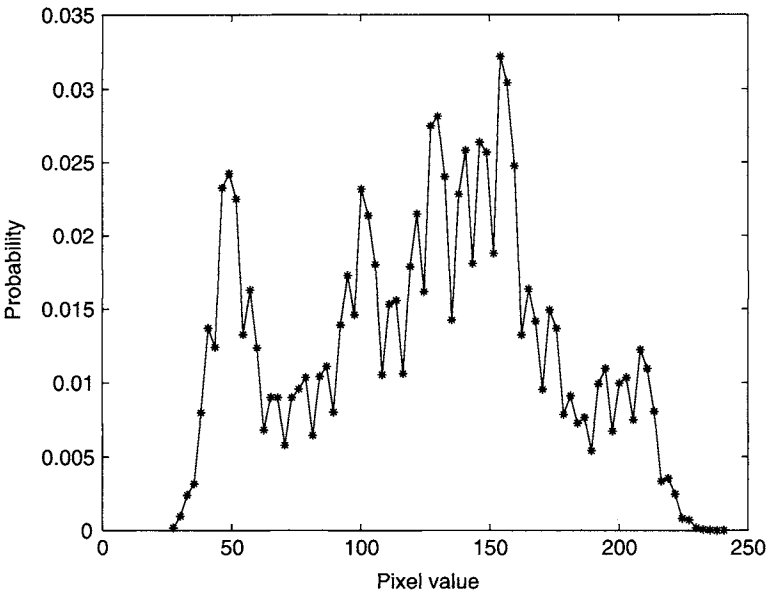

FIGURE 9.2
Probability mass function for the Lenna image.

of the image at varying spatial resolutions. The tools used in hierarchical coding methods are discussed in Section 9.2.2.

While predictive and transform-based methods have been used for lossless image compression, some other schemes have also been introduced to enhance the performance of these basic algorithms. These include special scanning techniques that determine the order in which the image is traversed in the coding process. Since scanning orders affect the definition of contexts, they could potentially have a significant effect on the compression performance. Scanning methods used in lossless image coding are discussed in Section 9.2.4.

9.2.1 Spatial Prediction

When there is significant memory between successive samples of a signal, it should be possible to predict the value of any given sample, with a reasonably high degree of accuracy, from some of the preceding samples. The number of samples used in the prediction is largely dependent on the extent of memory in the source and the complexity limits set on the prediction algorithm. Once the first sample (or samples, if more than one preceding sample is used in the prediction) is transmitted, and given that the decoder knows the prediction algorithm, we need transmit only the difference between the prediction and the actual pixel values for the following samples. This is because, given the first sample(s), the decoder will be able to generate its own prediction of the subsequent samples. Adding on the difference values received from the encoder will generate the exact replica of the original image. The difference between the actual signal and the predicted version is often called the *prediction residue* or *prediction error*. In this chapter we will use these terms interchangeably. If the prediction algorithm is reasonably good, then most of the values in the residue will be zero or very close to it. This in turn means that the pmf of the predicted signal will be peaky. The savings in bits comes from the fact that the prediction residues, which make up the bulk of the transmitted signal, can now be compressed well using entropy coding.

A simple predictor for an image is one that uses the previous pixel in the image as the prediction of the current pixel. Formally, if we denote the current pixel by I_n and the previous pixel by I_{n-1}, the prediction $\hat{I}_n$ of I_n is given by $\hat{I}_n = I_{n-1}$. In this case, the prediction error, e_n, is nothing but the difference between the adjacent pixels. Hence, $e_n = I_n - I_{n-1}$. Figure 9.3 shows the pmf of the difference between adjacent pixels for the Lenna image. Comparing this with Fig. 9.2, we can see that even a simple predictor like this can generate a pmf that is significantly skewed in comparison to that of the original image. With more sophisticated predictors, one can expect better gains. Typically, in lossless image compression, the number of samples used in the prediction is more than 1. The pixels used in the prediction are collectively referred to as a *context*. Figure 9.4 shows an example context along with some common notation used to refer to the individual pixels in it.

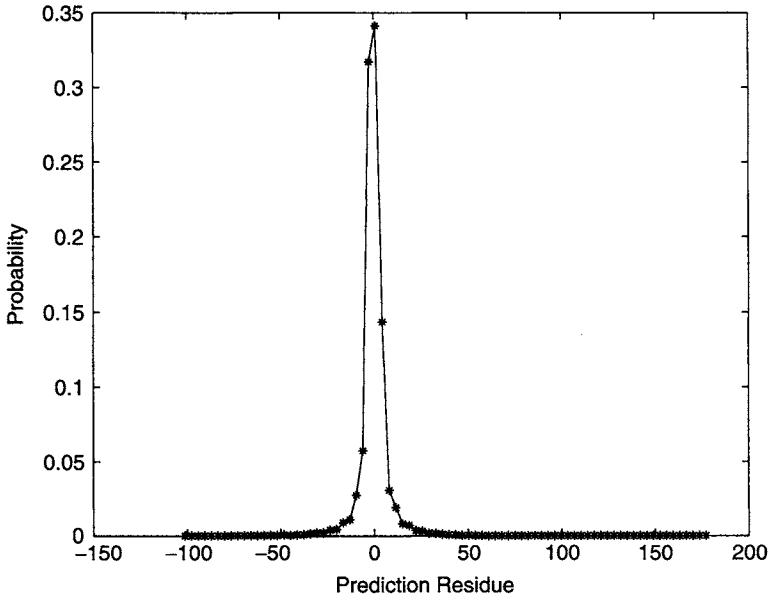

FIGURE 9.3
Probability mass function of the adjacent pixel differences for the Lenna image.

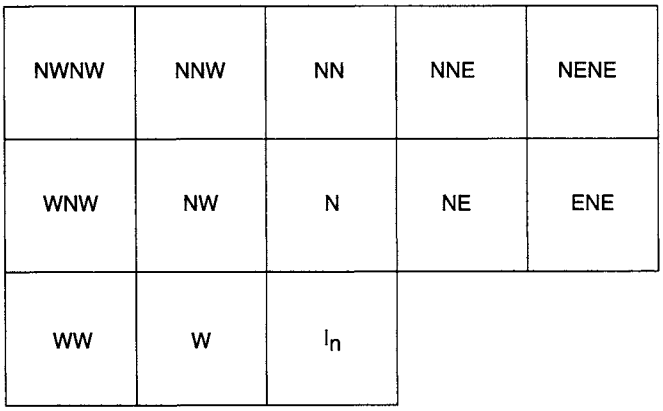

FIGURE 9.4
Example context.

Predictors are also classified into *linear* and *non-linear* predictors depending on whether the prediction for the current pixels is calculated as the linear combination of context pixels or not. When the prediction of the current pixel is calculated as the linear combination of the pixels in the context, the resulting predictor is considered to be a linear predictor. Mathematically, the linear prediction $\hat{I}_n$, of pixel I_n, can be represented by

$$\hat{I}_n = \sum_{i=1}^{K} \alpha_i I_{n-i}, \tag{9.1}$$

where $I_{n-i}, \forall i \in \{1, \ldots, K\}$ refers to pixels in the context and α_i are their corresponding weights in the linear combination. The design of a linear predictor then revolves around the selection of these weights. One approach to finding the weights would be to minimize the mean square prediction error. This, however, does not imply that the entropy of the prediction residue will be minimized.

Non-linear prediction algorithms have been generally less popular for lossless image compression. Some of the non-linear prediction algorithms use classification or learning techniques like neural networks and vector quantization. Rabbani and Dianat [2] used neural networks to predict pixels. Memon *et al.* [3] have looked at the problem of finding optimal non-linear predictors, in which they formulate it as a combinatorial optimization problem and propose some heuristic solutions.

While fixed predictors work well for stationary sources, predictors that *adapt* to the local characteristics of the source are more useful for images since images are essentially non-stationary signals. More details on these types of predictors are discussed in Section 9.3.

9.2.2 Hierarchical Prediction

Spatial predictive coding methods described in the previous section treat the image as a 1-D stream of symbols, whereas in reality, it is not. Considering the image as a 1-D stream imposes a causality constraint in defining a context for any pixel. This is because no pixel that appears after the current pixel in the scan order (see Section 9.2.4 for more on scanning techniques) can be a part of the current pixel's context. This problem can be avoided by using hierarchical or multiresolution prediction techniques that generate more than one representation of the image at different spatial resolutions. These representations can be constructed based on transform, subband, or pyramid coding.

In the basic pyramid coding scheme, the image is divided into small blocks and each block is assigned a representative value. This lower resolution version of the image is transmitted first. Other layers are added later to ultimately construct a lossless version of the image. A simple representation of the block would be its mean; however, more sophisticated schemes using transforms or subband filters are also in vogue. We do not attempt a comprehensive treatment of the theory of transform or wavelet coding here. Excellent sources for more information on this topic include [4–6].

Memon and Sayood [7] classify the hierarchical coding schemes as being either top-down or bottom-up depending on the way they are constructed, while Jung *et al.* [8] classify the existing pyramid-based techniques into either the *mean-based* methods or the *subsample-based* methods. When the average of the block is used as its representation, the pyramid coding scheme is considered to be mean based. In such a scheme a "mean pyramid" is generated where each point is the rounded-off average of non-overlapping image blocks. Once this mean pyramid is constructed, differences between the successive layers of this pyramid are transmitted. If, on the other hand, no

block averaging is done and the differences between successive layers of a pyramid are transmitted as is, the method is referred to as subsample based [8]. Examples of subsample-based pyramid construction can be found in [9–11] and instances of mean-based pyramid construction can be found in [12, 13].

9.2.3 Error Modeling

A key problem in prediction-based algorithms is the efficient representation of the error signal. One of the first prediction-based lossless image coders, the old JPEG standard [14], recommends that the prediction error be encoded using either Huffman codes or arithmetic codes. In the Huffman code option, the error is modeled as a sequence of independent identically distributed (i.i.d.) random variables. In general, however, even sophisticated predictors are unable to produce error signals that can be approximated by i.i.d. random variables. It is common to find some residual structure in the error signals. More sophisticated models are often designed to capture this structure.

The performance of entropy coders depends on the probability model that it uses for the data. If the probability model accurately describes the statistics of the source, then the entropy coder will be more efficient. This makes error or residue modeling [15] very crucial in the performance of the overall algorithm. Most of the error modeling techniques fall under the category of *context modeling* [16] as explained in [17] and applied in [15]. In this approach, the prediction error is encoded with respect to a conditioning state or context, arrived at from the values of previously predicted pixels. Essentially, the role of the error model in this framework is to provide an estimate of the pmf of the error, conditioned on some finite context. This conditional pmf can be estimated by counting the number of occurrences of the symbols within a context [18]. Alternatively, one could assume a model for the pmf of the error and calculate the appropriate parameters. Such an approach was taken in [15] where the pmf of the error was considered to be Laplacian and the variance was calculated. The context used to determine the pmf could be either the same as that used for prediction or different.

The context adaptive lossless image compression algorithm (to be discussed in greater detail in Section 9.3.1), also known as CALIC, chooses a different approach to modeling the prediction error. Here the conditional *expectation* ($E\{e|C\}$) of the error, e, within a context C, is estimated, rather than the conditional *probability*, $\Pr\{e|C\}$, of the error. The rationale for using this approach is based on the fact that the *conditional* mean (as opposed to the *unconditioned* mean) of the error signal is not necessarily zero and that the observed Laplacian distribution without conditioning on the context is a combination of different context-sensitive Laplacian distributions, with different means and variances [16]. Figure 9.5 illustrates this idea with two graphs. The graph labeled (b) represents a more realistic *probability distribution function* (pdf) for the prediction error and is a linear combination of several component Laplacian pdf's shown in the graph labeled (a). $E\{e|C\}$ is used to refine the prediction of the current pixel in CALIC and this process is termed *bias cancellation* by Memon and Wu [16].

9.2.4 Scanning Techniques

In most of the prediction-based lossless image compression techniques, the performance ultimately depends on an estimate of the conditional pmf for a pixel given its past. The past or causal context will depend on the way in which the image is "read" since the causality of the pixels will depend on the scanning method used. Although theoretically, the optimal bit rate at which the image is encoded will not depend on the scan order [19], this is true only if the context is allowed to be infinite. In practice, however, the context is finite and the scan order may be important.

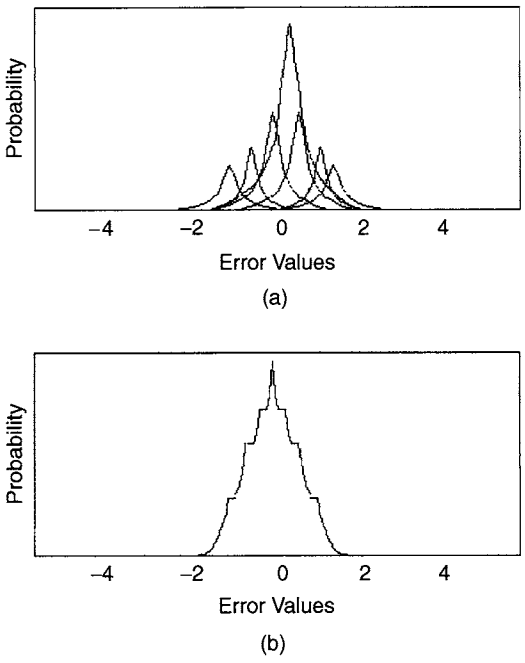

FIGURE 9.5
Distribution of prediction errors.

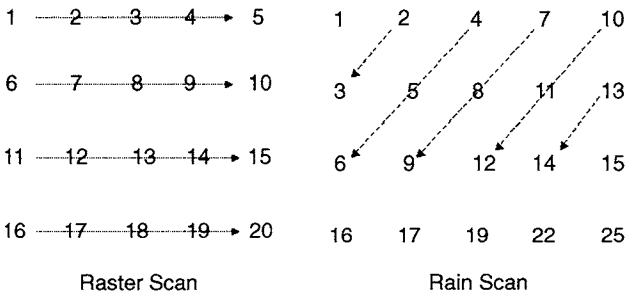

FIGURE 9.6
Scanning patterns.

Although most prediction algorithms use the raster scan order (where the pixels are read from left to right and top to bottom along the direction of the arrows in Fig. 9.6), Lempel and Ziv showed that universal coding of images can be accomplished by using a 1-D encoding method on the image scanned according to the discrete approximation of the Hilbert space-filling curves [20]. The Hilbert scan is shown in Fig. 9.7 for an 8×8 image block. Traversing the block in increasing order of the integer labels gives the Hilbert scan in this figure. Hilbert scanning has been used by many authors for its "clustering" or locality-preserving properties. For example, Perez *et al.* [21] use an adaptive arithmetic code for a one-dimensional stream obtained after Hilbert-scanning multispectral images. Very recently a systematic analysis of scanning patterns was conducted [19] and it was seen that for algorithms that use limited contexts (like most practical algorithms) the Hilbert scan may not lead to significant improvements over the raster scans. Hilbert scans

1	2	15	16	17	20	21	22
4	3	14	13	18	19	24	23
5	8	9	12	31	30	25	26
6	7	10	11	32	29	28	27
59	58	55	54	33	36	37	38
60	57	56	53	34	35	40	39
61	62	51	52	47	46	41	42
64	63	50	49	48	45	44	43

FIGURE 9.7
Hilbert scan.

were also shown to be not very advantageous in increasing run-length, in the work by Quin and Yanagisawa [22]. However, there are other scan orders proposed by certain authors especially as a feature to enhance the performance of their algorithm. One such ordering is the *rain* scan ordering proposed by Seeman *et al.* [23] (also shown in Fig. 9.6).

9.3 PREDICTION FOR LOSSLESS IMAGE COMPRESSION

As mentioned earlier, while static (unchanging) prediction is a good tool for effectively decorrelating a stationary signal, this technique will have to be used with some enhancements when dealing with a non-stationary source. Non-stationary sources like images are characterized by abruptly varying statistics. In order to keep up with this change it would be desirable to make the predictors *adaptive* to local statistics of the image. The adaptive predictors used in lossless image compression can be broadly classified as *switched predictors* or *combined predictors*.

The first class encompasses all techniques where a single subpredictor is chosen from a set. For linear predictors, one might also consider changing the weights of the context pixels in the linear combination according to the local characteristics. This can also be thought of as switched prediction, where the set of subpredictors is calculated somewhat on the fly. In the second category, namely, combined prediction, we do not choose a *single* subpredictor from a set, but rather combine a subset of subpredictors from the entire collection. Figure 9.8 shows a further subclassification of these categories.

9.3.1 Switched Predictors

Linear predictors can be made adaptive by changing the weights in the linear combination of causal neighboring pixels (α_is in Eq. (9.1)). Alternatively, it is possible to select one predictor from a set of subpredictors that may even be non-linear. Switched prediction encompasses both of the above categories; in fact, the weight-changing strategy may itself be considered a technique of choosing one subpredictor from a repertoire of subpredictors.

The parameters that influence the choice of subpredictor could either be based on the performance of the subpredictor on a set of causal neighbors (performance-based selection) or be based on some local characteristic of the image (local characteristics-based selection).

Sayood and Anderson [24] proposed a switched prediction scheme where the switching is done based on the prediction error performance of the subpredictors. Another simple and effective

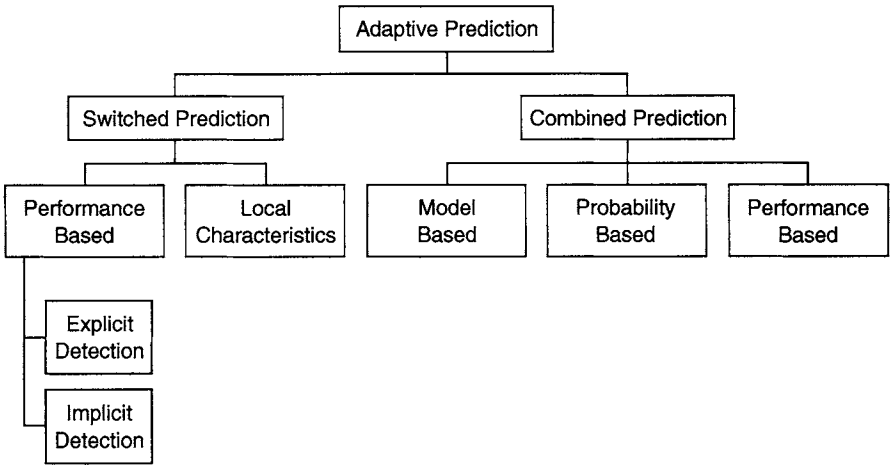

FIGURE 9.8
Classification of adaptive prediction schemes.

adaptive predictor was suggested in [25], where the median prediction of a set of subpredictors is calculated and this is used as the actual prediction. A more elaborate prediction scheme that involves the prediction errors for a causal neighborhood of the current pixel was developed in [26]; however, Memon and Sayood [7] contend that for lossless compression this technique does not offer much improvement over the median adaptive predictor.

An image is made up of different regions like edges, smooth surfaces, and textures, all of which have very different characteristics. Predicting pixels that fall in any one of these categories from pixels that belong to other categories can be inefficient. In fact, it may be necessary to change not only the context of the subpredictor, but also the subpredictor's structure according to the local characteristics of the image. Some switched prediction techniques approach this problem by explicitly detecting the presence of edges or gradients and then changing the subpredictor accordingly, while others "detect" these different regions implicitly. Accordingly, the switched predictors based on local characteristics of the image are further classified into those that explicitly detect features and those that implicitly "learn" them (see Fig. 9.8). Some performance-based switched predictors, like the ALCM [27] and JSLUG [28], do not detect edges either explicitly or implicity. The ALCM and JSLUG include an adaptive predictor that uses a weighted combination of 5 neighborhood pixels. Initially all pixels are assigned equal weights. They are then modified on the fly depending on the prediction error. If the prediction value was higher than the actual pixel value, then the weight of the largest pixel is decremented by $\frac{1}{256}$ and the weight of the smallest pixel in the neighborhood is incremented by the same amount. If the prediction is too low, the exact opposite scheme is used. In the case of ties in the weights of highest (or lowest) pixels, the ties are broken according to the following priority scheme. The pixel labeled 1 in Fig. 9.9 is changed with the highest priority and the pixel labeled 5 with least priority.

9.3.1.1 Explicit Detection-Based Predictors

Explicit detection-based predictors detect edges, gradients, or texture regions in images using simple thresholding and some measure of activity in the image. All of the algorithms covered in this section are more recent and were developed in the wake of the call for proposal [29] issued by the International Standards Organization (ISO) in 1994. A comprehensive comparison [16] of all the proposals submitted to the ISO was undertaken by Memon and Wu and some of those algorithms are described in greater detail here.

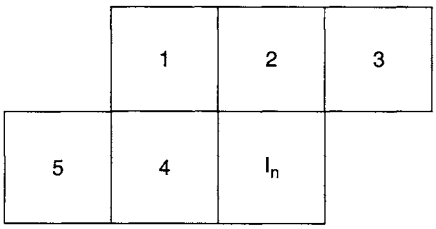

FIGURE 9.9
Priority scheme for resolving ties.

The median edge detection-based (MED) predictor [30] adapts to the local edges which it detects through a simple algorithm. The N, W, and NW pixels (see Fig. 9.4) are used to detect the presence and orientation of a local edge. If a local vertical edge is detected, then the N pixel is used as the prediction of the current pixel. If a horizontal edge is detected, then the W pixel is used for prediction, and if no edge is detected, then the planar predictor is used, where the prediction $\hat{I}_n$ of the pixel I_n is set to $N + W - NW$.

The MED predictor was used in the low-complexity lossless compression for images (LOCO-I) [30, 31], which was Hewlett-Packard's submission to the standardization committee. This algorithm was adopted in the JPEG-LS standard for lossless and near lossless compression.

The CALIC [32] algorithm detects edges using local gradients in the image. CALIC uses the gradient adaptive predictor (GAP), which changes the weights of the context pixels according to the local gradients in the image. The vertical and horizontal gradients are estimated according to the following equations:

$$d_v = |W - NW| + |N - NW| + |N - NE| \tag{9.2}$$

$$d_h = |W - NW| + |N - NN| + |NE - NNE|. \tag{9.3}$$

Once the gradients are estimated, the presence or absence of edges, as well as their orientation, is determined by comparing the difference between the vertical and the horizontal gradients to some experimentally determined thresholds. Then the weights of the pixels used in the predictor are determined. More details can be found in [16]; the CALIC algorithm also uses the bias-cancellation procedure mentioned in Section 9.2.3 as an error feedback mechanism to fine-tune the prediction.

Other prediction schemes in this category include CLARA [33], Mitsubishi's proposal to the standardization effort. CLARA used texture information as well as edge/gradient detection in its prediction.

The study by Memon and Wu [16] found that among the prediction-based algorithms submitted to the ISO, the MED, GAP, and ALCM predictors performed alike in terms of the zeroth order entropy of the prediction error; however, the MED and GAP predictors are computationally less complex than the ALCM predictor.

9.3.1.2 Learning-Based Switched Predictors

Recently, a switched predictor algorithm [34] was designed that uses a clustering-based approach to classify the local characteristics of the image and a gradient descent algorithm to refine an initial set of weights. The algorithm first defines a causal window of pixels for the current pixel and classifies these pixels and their contexts into a given number of clusters. A centroid is then defined for each cluster. Then the current pixel and its context are classified into one of these

clusters (with centroid C_k, say) and then a set of weights ($P_i = \{w_0, w_1, \ldots w_r\}$, where r is the number of pixels in C_k) are picked that minimize the prediction error for the current pixel, using the pixels in C_k. Finally, the gradient descent algorithm is applied to the pixels in C_k so as to refine the set of weights, P_i. The prediction step in this algorithm is rather complex; however, the algorithm performs as well as CALIC for most images and better than CALIC for some [34]. Their experiments suggest that with more sophisticated prediction error encoding, the algorithm can be competitive with CALIC.

The edge-directed prediction algorithm developed by Li and Orchard [35] is another such predictor where the switching is implicit and the local characteristics of the image are learned. This algorithm does not explicitly detect edges, although it uses the fact that the edges are fundamentally different from the non-edge areas. Therefore, the ideal predictor would align the support of the local pmf in the edge area along the edge itself and would thus have a 1-D support. This is intuitively clear if one considers that the best context for any edge pixel would be found ideally along the edge and not in the area away from it. The local pmf of the rest of the image (non-edge parts) will have a 2-D support, since there is no reason to choose context pixels from any one direction over the other.

The best linear predictor for an Nth order Markov source $\{x(i)\}$ is given by

$$\hat{x}(i) = E(x(i)|\vec{x}) = \vec{\alpha}\vec{x}^T, \quad \vec{r}_x R_{xx}^{-1}, \tag{9.4}$$

where $\vec{x} = [x(i - n_1) \ldots x(i - n_N)]$, $\vec{r}_x = [r_1, r_2, \ldots, r_N]$, $r_k = \text{Cov}\{x(i)x(i - n_k)\}, \forall\, k \in \{1, \ldots, N\}$ and $R_{xx} = [R_{jk}]$, $R_{jk} = \text{Cov}\{x(i - n_j)x(i - n_k)\}$, $\forall\, k, j \in \{1, \ldots, N\}$. So, determining the optimal prediction would involve estimating the covariance matrix, R_{xx}, of the signal. When the signal is not stationary, as is the case with images, the above formulation will have to be made adaptive. One way to do this within the above framework is to assume that the image is locally stationary and estimate the covariance matrix of the image based on some causal neighborhood of the current pixel. The estimation of the covariance matrix is done using a least-squares approach in this algorithm by Li and Orchard.

Finally, the predictor coefficients can also be determined using the least-mean-square (LMS) algorithm as in [36] and in one of the subpredictors in the combined predictors algorithm proposed in [37] (see Section 9.3.2).

9.3.2 Combined Predictors

Predictors can also be viewed from a modeling perspective, where each predictor is considered to be a model for the current pixel. A model is usually constructed based on some assumptions about the modeled data. If these assumptions are correct, then the model is said to be "good" for the data. If it is not, then the model will not produce a good prediction of the data. For non-stationary sources, it is difficult to know to what extent the assumptions made by any model are correct. Hence there is an uncertainty about the model to be used. Some authors [37, 38] contend that this uncertainty warrants the use of combined predictors rather than switched predictors. In combined prediction, more than one predictor is used to predict the current pixels. These predictions are then combined to form the final predicted value.

One way of combining subpredictors is to form a linear combination of the prediction values that each of the subpredictors produces. The design issue is then to choose the coefficients of this linear combination. While some authors view the problem as one of modeling (model-based blending), some others prefer a more intuitive approach like penalizing the subpredictors by a term proportional to the prediction error (performance-based blending). Some other authors have also suggested that the *probability distributions* be combined rather than the prediction values (probability blending).

9.3.2.1 Performance-Based Blending of Predictors

Seeman and Tischer proposed a prediction blending algorithm [39] where the coefficients in the linear combination can be used to penalize the subpredictors that result in large prediction errors. Let I_n be the pixel to be predicted. Let $p_j(n-k)$; $j \in \{1, \ldots, M\}$ be the jth subpredictor for pixel I_{n-k}. The penalty term for the jth subpredictor is calculated as follows:

$$G_j = \sum_{k=1}^{N} |I_{n-k} - p_j(n-k)|. \tag{9.5}$$

The prediction for the current pixel is then given by

$$I_n = \frac{1}{D} \sum_{j=1}^{M} \frac{p_j(n)}{G_j}, \tag{9.6}$$

where D, the normalization factor, is given by $D = \sum_{j=1}^{M} 1/G_j$. G_j in Eq. (9.5) is computed over a small neighborhood (consisting of M pixels) of the current pixel. This algorithm was tested extensively [39] and it was seen to outperform both MED and GAP in terms of the entropy of the error signal. When coupled with context-based error feedback and entropy coding, it was seen to perform as well as or better than LOCO and almost as well as CALIC for natural images.

Seeman and Tischer then extended their algorithm to another algorithm called the history-based blending of predictors (see Ref. [23]). This subpredictor combination algorithm uses more complex non-linear subpredictors and the mean absolute error (MAE) criterion for blending the predictors rather than the mean squared error (MSE). This is because of the robustness offered by the MAE over the MSE criterion [40]. This algorithm does as well as the CALIC and the LOCO algorithms, albeit at increased complexity [23]. Part of the success of this algorithm has been attributed to the rain order scanning discussed in Section 9.2.4.

In [41, 42], Aiazzi *et al.* introduced a clustering-based combined prediction algorithm. Here, the image is first split into several blocks and the minimum mean square error linear predictor is calculated for each block. These are then classified into M prototype subpredictors. The image is then traversed in raster scan order (Fig. 9.6) and the MSE is calculated for each pixel using each subpredictor. The M subpredictors are then combined using weights proportional to the inverse of their MSEs. More details on how the clustering is done and the actual expression for the weights can be found in [41].

9.3.2.2 Model-Based Blending

As noted earlier, predictors can be viewed as models for a particular pixel. In the Bayesian approach to combining subpredictors, each subpredictor is considered to be a model and a risk is associated with picking each of these models. Bayesian model averaging (BMA) is a coherent mechanism to combine these models. According to BMA, given the data D and a group of models M_k, the Bayesian estimate θ_B of a parameter θ is determined by

$$\theta_B = E[\theta|D] = \sum_{k=1}^{K} \theta_k \Pr(M_k|D), \tag{9.7}$$

where $\theta_k = E[\theta|M_k]$. It has been demonstrated that BMA provides better predictive ability than choosing any single model [43].

Lee [37] recently proposed a linear combination of subpredictors where the coefficients in the linear combination were determined using a Bayesian approach. In this algorithm five different subpredictors are used, four of which are simple predictors using the previous pixel as the prediction of the current pixel. The fifth subpredictor is adaptive, and the weights of the causal neighbors

in the linear combination are determined using a LMS-based algorithm. The prediction errors of the subpredictors are modeled as Laplacian distributions and the *a posteriori* probability of the prediction error is used as the combination coefficient in the linear blending of predictors. The performance of this algorithm was found to be comparable to that of CALIC [37].

Deng and Ye [44] used a Lagrange multiplier method in the linear combination of subpredictors. If $p_j(n)$ is the prediction of the nth pixel using the jth subpredictor, then $e_j(n)$ is the corresponding prediction error. The linear combination of subpredictors for the nth pixel is given by

$$P(n) = \sum_{j=1}^{M} \alpha_j(n) p_j(n), \tag{9.8}$$

where $\alpha_j(n)$ are the coefficients in the linear combination of predictors. If we denote the overall prediction error for the combined predictor by $e(i)$, then $e(i) = I(i) - P(i)$, where $I(i)$ is the current pixel and $P(i)$ is its prediction using the combined predictor. The algorithm proposed by Deng and Ye uses the Lagragian multiplier method

$$F = \sum_i e^2(i) + \beta\left(1 - \sum_{j=1}^{M} \alpha_j(n)\right), \tag{9.9}$$

to minimize the mean square error of the predictor over a training block of pixels with the condition that the $\alpha_j(n)$s (Eq. (9.8)) sum to 1. The coefficients of the linear combination are found by solving the following set of equations:

$$\frac{\partial F}{\partial \alpha_j(n)} = 0 \quad j = 1, 2, \ldots, M \tag{9.10}$$

$$\frac{\partial F}{\partial \beta} = 0 \tag{9.11}$$

How this algorithm fits in the Bayesian model averaging framework is described in greater detail in [38].

9.3.2.3 Probability Domain Blending

In an entirely different approach to blending subpredictors, Meyer and Tischer proposed an algorithm [45, 46] where the blending took place in the probability domain rather than the spatial domain. In this two-pass algorithm, the first part consists of selecting several parameters (weights of the individual subpredictors) for the encoding in such a way as to produce the shortest encoded stream. The encoding per se takes place in the second pass. Because of the two-pass nature of this algorithm, the resulting compressed image consists of a header followed by the actual encoded image. Apart from the usual pixel predictor, this algorithm uses two other kinds of predictors: one that predicts the magnitude of *prediction error* of the subpredictor based on the prediction errors of a set of causal neighbors and another that predicts how well the pixel predictor will perform on a certain pixel given its performance in a causal neighborhood.

Since the other combined predictors discussed so far produce only a single prediction value, there will be only a single pdf used to ultimately code the pixel. This pdf will be centered around the resulting predicted value. Thus, it will not be possible to model more complex pdf's (like bimodals) if this pdf is itself not complex. So, these methods could potentially suffer along edges. This algorithm gets around this problem by first calculating the pdf of the current pixel using *each* of the subpredictors and then blending the probability density functions rather than the prediction values. More details on the probability model used in this method and the combination of pdf's can be found in [46].

9.4 HIERARCHICAL LOSSLESS IMAGE CODING

As noted in Section 9.2.2, spatial prediction techniques impose a causal structure on the image which is essentially artificial. Hierarchical prediction schemes avoid imposing such a structure on the image by splitting an image into different *layers*, each of which represents the entire image but at different spatial resolutions. One of the first hierarchical coding algorithms to be developed was *hierarchical interpolation* (HINT) [47]. In the HINT algorithm, the pixels marked Δ in Fig. 9.10 are transmitted first using lossless DPCM, then the pixels marked $\circ$ are predicted from the first set and the prediction error is transmitted. Next, the pixels marked X are predicted from the $\circ$ and Δ pixels and the prediction error is transmitted, and finally, the pixels labeled $\star$ and $\bullet$ are predicted from their neighbors (now known) and the error is transmitted. Linney and Gregson [48] proposed a modified HINT scheme where the subsampling rate was varied according to the local frequency content of the image. Other enhancements to the HINT algorithm can be found in [49, 50].

Pyramidal or hierarchical schemes can be created by using both subband and wavelet transforms. In their pioneering work, Burt and Adelson [51] developed the Laplacian pyramidal decomposition of images. In their scheme, the image I_0 is lowpass-filtered to obtain G_0. The difference between I_0 and G_0 is saved as L_0, and then G_0 is down-sampled by a factor of 2 along the vertical and horizontal directions to yield I_1. The process is then applied to the down-sampled version I_1, to generate G_1, L_1, and I_2. A third level is then generated along the same lines, to yield G_2, L_2, and I_3. The subbands I_3, L_2, L_1, and L_0 are transmitted. The decoder recovers the signal by successive upsampling, interpolation, and addition of difference images. The general pyramid coding scheme for two stages is shown in Fig. 9.11. Although down-sampling in both directions is represented by a *single* block in this figure, it must be remembered that after each such down-sampling, the total number of pixels is reduced to a quarter of the original value.

In the basic pyramid structure, the total number of data values that need to be transmitted for an $N \times N$ image is greater than N^2. Hence there is an expansion in the number of points that need to be transmitted. Despite this increase, it is possible to encode the mean-residue pyramid using fewer bits than needed to entropy code the image. It is also possible to decrease this expansion in several ways. One of the first schemes to achieve this goal was the *bin-tree* data structure developed to represent an image, along with a way to map pixel pairs to *composite-differentiator* pairs proposed in [52]. Another popular approach that does not cause data expansion is the

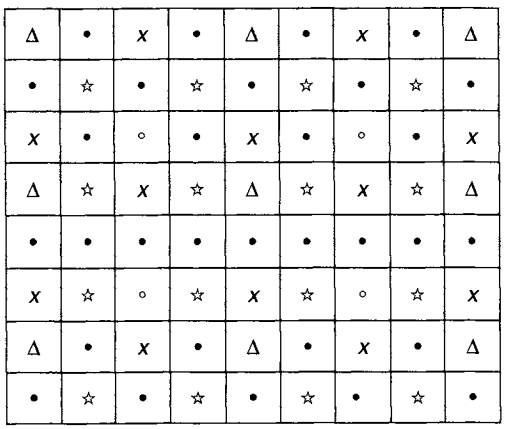

FIGURE 9.10
Hierarchical interpolation scheme for decorrelation.

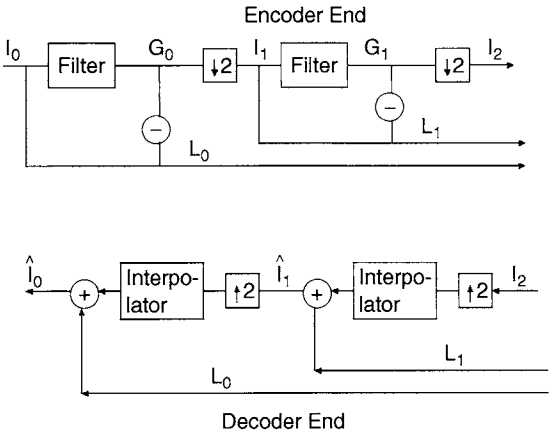

FIGURE 9.11
A generic pyramid coding scheme.

reduced-pyramid structure proposed by Wang and Goldberg [53]. A reduced *Laplacian* pyramid structure was proposed in [9]. Other reduced-pyramid structures based on transforms were presented in [13, 54–56].

Jung *et al.* proposed a rounding transform [8] that could be used in a lossless pyramid structured coding. This transform maps an integer vector to another by using weighted average and difference filters, followed by a rounding operation, and it generalizes some of the previous pyramid-based schemes like those found in [12, 13, 53]. It was then extended to an overlapping rounding transform [57], which is defined as a two-port FIR filtering system with a pair of rounding operations. Very recently, a method for optimal construction of a class of reduced pyramids was presented in [58], where the interpolation synthesis postfilters are selected to minimize the error variance at each level of the pyramid.

Although transform coding has been very useful in the context of lossy signal compression, there is one significant drawback to using this technique directly for lossless image compression: the transform coefficients are real numbers. Representing real numbers efficiently and without any loss is tricky. Although dyadic representation of the numbers with rescaling can be used, this causes an increase in the dynamic range. In [59–61] a new technique was described, called *lifting*, which allows the transformation of any perfect reconstruction filter to an integer equivalent. Any perfect reconstruction filter bank structure can be transformed to a lifting structure (LS) [59]. For the LS, the signal is split into two streams consisting of the even indexed and odd indexed samples, respectively. One of the streams is passed through a lifting filter and added onto the other stream. The role of the two streams can be reversed and the lifting steps reapplied. Figure 9.12 shows a schematic representation of the lifting steps. After K stages, one of the streams corresponds to the high-pass signal, $d^{(n)}$, and the other corresponds to the low-pass signal, $s^{(n)}$. Lifting steps that use the low-pass signals are called prediction steps and those that use the high-pass signal are called update steps. In Fig. 9.12, the steps that use $H_i(z)$, where i is even, are called the prediction steps. The inverse filter is given by $H'_k(z) = -H_k(z)$. The interesting point about the LS construction is that one could replace the lifting filters with a non-linear step and still have perfect reconstruction. So, introducing a rounding-off operation in the steps, as shown in Fig. 9.13, creates an integer wavelet transform (IWT) where integer inputs produce integer outputs [62]. IWT was applied to lossless image compression in [63].

Several researchers have studied IWT for lossy and lossless image compression [64, 65]. A theoretical framework to quantify the distortion caused by the use of IWT for lossy image

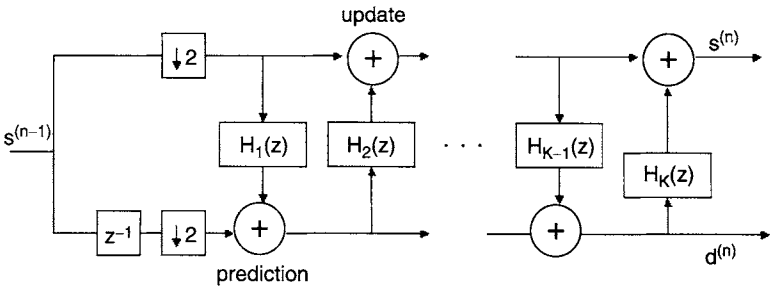

FIGURE 9.12
Basic lifting-based wavelet decomposition.

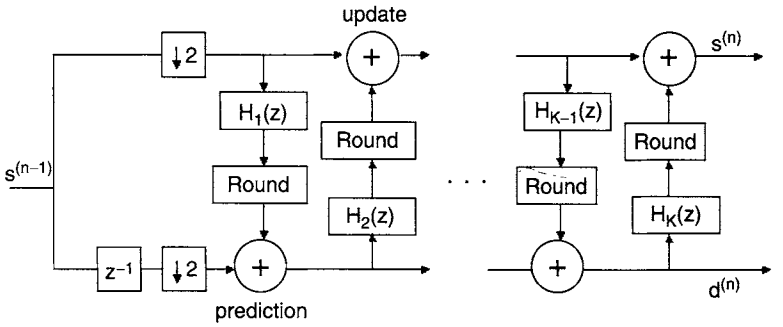

FIGURE 9.13
IWT based on the lifting scheme.

compression was presented in [66]. Boulgouris *et al.* [67] constructed optimal filters in the sense of minimizing the variance of the prediction error of a wide sense stationary signal of a lifting in the general $n-$dimensional case. Cheung *et al.* have designed a filter for use with the lifting scheme based on the box and slope multiscaling system [68]. Other attempts to address the issue of finding efficient filters for use with a lifting scheme include [69], which considers the problem of finding high-performance factorizations of the wavelet filters. A two-layered approach, based on the use of wavelet *packets* in the reversible integer format, was presented in [70], which makes the scheme locally adaptive. Since the codecs based on the discrete wavelet transforms can be used with IWT, there has also been interest in using a similar structure for lossless coding [71–74].

9.5 CONCLUSIONS

An image is characterized by a significant amount of redundancy in the sense that the adjacent pixels in an image are highly correlated. Making use of this correlation forms the crux of the lossless image compression techniques. One of the ways to use this correlation is to predict a pixel from its neighbors. This can be done in the spatial domain using a causal context for the pixel or it can be done hierarchically using pyramid, subband, or transform coding methods. The prediction methods involve transmitting the prediction errors efficiently, usually involving good statistical models for the errors. Lossless compression of images can also be affected by

the scanning techniques that are used in defining the context. This chapter discussed different strategies that exploit the correlation in the image and various error modeling techniques and scanning patterns used in lossless image compression.

9.6 REFERENCES

1. Deleted at proof.
2. Rabbani, M., and S. Dianat, 1993. Differential pulse code modulation compression of images using artificial neural networks. In *SPIE, Image and Video Processing*, pp. 197–203.
3. Memon, N., S. Ray, and K. Sayood, 1994. Differential lossless encoding of images using nonlinear prediction techniques. In *International Conference on Image Processing*, pp. 841–845.
4. Sayood, K., 2000. *Introduction to Data Compression*, 2nd ed. Multimedia Information Systems, San Francisco, CA, and Morgan Kauffman, San Mateo, CA.
5. Akansu, A., and M. Smith (Eds.), 1996. *Subband and Wavelet Transforms Designs and Applications*. Kluwer Academic, Dordrecht/Norwell, MA.
6. Burrus, C., R. Gopinath, and H. Guo, 1998. *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, New York.
7. Memon, N., and K. Sayood, 1995. Lossless image compression—A comparative study. *SPIE Proceedings, Still Image Compression*, Vol. 2418, pp. 8–20.
8. Jung, H.-Y., T.-Y. Choi, and R. Prost, 1998. Rounding transform and its application to lossless pyramid structured coding. *IEEE Transactions on Image Processing*, Vol. 7, pp. 234–237, February 1998.
9. Aiazi, B., L. Alparone, and S. Barantoni, 1996. A reduced Laplacian pyramid for lossless and progressive image communication. *IEEE Transactions on Communications*, Vol. 44, pp. 18–23, January 1996.
10. Houlding, D., and J. Vaisey, 1995. Low entropy image pyramids for efficient lossless coding. *IEEE Transactions on Image Processing*, Vol. 4, pp. 1150–1153, August 1995.
11. Roos, P., M. Viergever, M. V. Dijke, and J. Peters, 1998. Reversible intraframe compression of medical images. *IEEE Transactions on Medical Imaging*, Vol. 4, pp. 328–336.
12. Blume, H., and A. Fand, 1989. Reversible and irreversible image data compression using the s-transform and Lempel Ziv coding. *SPIE Medical Imaging III: Image Capture and Display*, Vol. 1091, pp. 2–18.
13. Kim, W., P. Balsara, D. H., III, and J. Park, 1995. Hierarchy embedded differential images for progressive transmission using lossless compression. *IEEE Circuits and Systems for Video Technology*, Vol. 5, pp. 1–13, February 1995.
14. Pennebaker, W., and J. Mitchell, 1993. *JPEG Still Image Compression Standard*. Van Norstrand Reinhold, New York.
15. Howard, P., and J. Vitter, 1992. Error modeling for hierarchical lossless image compression. In *Data Compression Conference*, pp. 269–278.
16. Memon, N., and X. Wu, 1998. Recent developments in lossless image compression. *The Computer Journal*, Vol. 40, pp. 117–126, June 1998.
17. Rissanen, J., and G. Langdon, 1981. Universal modeling and coding. *IEEE Transactions on Information Theory*, Vol. 1, pp. 12–22, January 1981.
18. Todd, S., G. Langdon, and J. Rissanen, 1985. Paramter reduction and context selection for compression of gray scale images. *IBM Journal of Research and Development*, Vol. 29, No. 2, pp. 188–193.
19. Memon, N., D. L. Neuhoff, and S. Shende, 2000. An analysis of some common scanning techniques for lossless image compression. *IEEE Transactions on Image Processing*, Vol. 9, pp. 1837–1848, November 2000.
20. Lempel and Ziv, 1986. Compression of two-dimensional data. *IEEE Transactions on Information Theory*, Vol. 32, No. 1, pp. 1–8.
21. Perez, A., S. Kamata, and E. Kawaguchi, 1991. Arithmetic coding model for compression of landsat images. *Visual Communications and Image Processing*, Vol. SPIE 1605, pp. 879–884.
22. Quin, A., and Y. Yanagisawa, 1989. On data compaction of scanning curves. *Computer Journal*, Vol. 32, No. 6, pp. 563–566.

23. Seeman, T., P. Tischer, and B. Meyer, 1997. History-based blending of image subpredictors. In *Proceedings of the Picture Coding Symposium*, Berlin, Germany, pp. 147–151.
24. Sayood, K., and K. Anderson, 1992. A differential lossless image compression algorithm. *IEEE Transactions on Signal Processing*, Vol. 40, No. 1, pp. 236–241.
25. Martucci, S., 1990. Reversible compression of HDTV images using meidan adaptive prediction and arithemtic coding. In *IEEE, International Symposium on Circuits and Systems*, pp. 1310–1313.
26. Zschunke, W., 1977. DPCM picture coding with adaptive prediction. *IEEE Transactions on Communications*, Vol. 25, No. 11, pp. 1295–1302.
27. Speck, D., 1996. Fast robust adaptation of predictor weights from min/max neighbouring pixels for minimum conditional entropy. In *Twenty-ninth Asilomar Conference on Signals, Systems and Computers, 1997*, Asilomar, pp. 234–238.
28. Langdon, G., D. Speck, C. Haidinyak, and S. Macy, 1995. Contribution to JTC 1.29.12: JSLUG, ISO Working Document ISO/IEC JTC1/SC29/WG1 N199.
29. ISO/IEC JTC 1/SC 29/WG1, 1994. Call for Contributions—Lossless Compression of Continuous Tone Still Pictures. ISO Working document ISO/IEC JTC1/SC29/WG1/N196.
30. Weinberger, M., G. Seroussi, and G. Sapiro, 1995. LOCO-I: A Low Complexity Lossless Image Compression Algorithm, ISO Working document ISO/IEC JTC1/SC29/WG1 N281, August 1995.
31. Weinberger, M., G. Seroussi, and G. Sapiro, 2000. LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Transactions on Image Processing*, Vol. 9, pp. 1309–1324, August 2000.
32. Wu, X., N. Memon, and K. Sayood, 1995. A Context Based, Adaptive, Lossless/Near-Lossless Image Coding Scheme for Continous-Tone Images, ISO Working document ISO/IEC JTC1/SC29/WG1/N256.
33. Ueno, I., and F. Ono, 1995. CLARA: Continuous-Tone Lossless Coding with Edge Analysis and Range Amplitude Detection, ISO Working document ISO/IEC JTC1/SC29/WG1/N197.
34. Motta, G., J. Storer, and B. Carpentieri, 2000. Lossless image coding via adaptive linear prediction and classification. *Proceedings of the IEEE*, Vol. 88, pp. 1790–1796, November 2000.
35. Li, X., and M. T. Orchard, 1998. Edge directed prediction for lossless compression of natural images. In *IEEE International Symposium on Circuits and Systems*, pp. 58–62.
36. Chung, Y., and M. Kanefsky, 1992. On 2-D recursive LMS algorithm using the ARMA prediction for ADPCM encoding of images. *IEEE Transactions on Image Processing*, Vol. 1, pp. 416–442.
37. Lee, W., 1999. Edge adaptive prediction for lossless image coding. In *Data Compression Conference*, pp. 483–490.
38. Deng, G., H. Ye, and L. Cahill, 2000. Adaptive combination of linear predictors for lossless image compression. *IEE Proceedings—Scientific Measurements and Technology*, Vol. 147, pp. 414–419, November 2000.
39. Seemann, T., and P. Tischer, 1997. Generalized Locally Adaptive DPCM, Technical Report 97/301, Monash University, Australia.
40. Tischer, P., 1994. Optimal Predictors for image Compression, Technical Report 94/189, Monash University, Australia.
41. Aiazi, B., S. Baronti, and L. Alparone, 2000. Near lossless compression using relaxation labeled prediction. In *International Conference on Image Processing*, Vancouver, British Coloumbia, Canada, pp. 148–151.
42. Aiazi, B., S. Baronti, and L. Alparone, 1999. Lossless image compression based on an enhanced fuzzy regression prediction. In *International Conference on Image Processing*, Berlin, Germany, pp. 435–439.
43. Hoeting, J., D. Madigan, A. Raftery, and C. Volinsky, 1998. Bayesian model averaging: A tutorial, Technical Report 9814, Department of Statistics, Colorado State University, Fort Collins.
44. Deng, G., and H. Ye, 1999. Lossless image compression using adaptive predictor combination, symbol mapping and context filtering. In *International Conference on Image Processing*, Kobe, Japan, Vol. 4, pp. 63–67.
45. Meyer, B., and P. E. Tischer, 1997. TMW—A new method for near lossless compression of greyscale images. In *Picture Coding Symposium*, Berlin, Germany, pp. 533–538.
46. Meyer, B., and P. Tischer, 1998. Extending tmw for lossless image compression. In *Data Compression Conference*, pp. 458–470.

47. Endoh, T., and Y. Yamazaki, 1984. Progressive coding scheme for interactive image communications. In *Global Communications Conference*, Vol. 3, pp. 1426–1433.
48. Linney, N., and P. Gregson, 1994. HINT based multiresolution image compression. In *Canadian Conference on Electrical and Computer Engineering*, Vol. 2, pp. 580–583.
49. Ramabadran, T., and K. Chen, 1992. The use of contextual information in the reversible compression of medical images. *IEEE Transactions on Medical Imaging*, Vol. 11, No. 2, pp. 185–195.
50. Howard, P., and J. Vitter, 1991. New methods for lossless image compression using arithmetic coding. In *Data Compression Conference*, pp. 257–266.
51. Burt, P., and E. Adelson, 1983. Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, Vol. Com-31, pp. 532–540, April 1983.
52. Knowlton, K., 1980. Progressive transmission of grey-scale and binary pictures by simple, efficient and lossless encoding schemes. *Proceedings of the IEEE*, Vol. 68, No. 7, pp. 885–896.
53. Wang, L., and M. Goldberg, 1989. Reduced-difference pyramid: A data structure for progressive image transmission. *Optical Engineering*, Vol. 28, pp. 708–716, July 1989.
54. Rabbani, M., and P. Jones, 1991. *Digital Image Compression Techniques*, Vol. TT7, Tutorial Texts Series. *Int. Soc.* Opt. Eng., Bellingham, WA.
55. Said, A., and W. Pearlman, 1993. Reversible image compression via multiresolution representation and predictive coding. In *Visual Communications and Signal Processing*, pp. 664–673.
56. Zandi, A., J. Allen, E. Schwartz, and M. Boliek, 1995. CREW: Compression by reversible embeddded wavelets. In *Data Compression Conference*, pp. 664–673.
57. Jung, H., and R. Prost, 1997. Rounding transform based approach for lossless subband coding. In *International Conference on Image Processing*, Santa Barbara, CA, Vol. 2, pp. 271–278.
58. Tzovaras, D., and M. Strintzis, 2000. Optimal construction of reduced pyramids for lossless and progressive image coding. *IEEE Transactions on Circuits and Systems.II. Analog and Digital Signal Processing*, Vol. 47, pp. 332–348, April 2000.
59. Sweldens, W., 1996. The lifting scheme: A custom-design construction of biorthgonal wavelets. *Journal of Applied and Computer Harmonic Analysis*, Vol. 3, pp. 186–200.
60. Daubechies, I., and W. Sweldens, 1996. Factoring Wavelet Transforms into Lifting Steps, Technical Report, Bell Laboratories, Lucent Technologies.
61. Daubechies, I., and W. Sweldens, 1998. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, Vol. 4, No. 3, pp. 245–267.
62. Calderbank, R., I. Daubecheis, W. Sweldens, and B-L.Yeo, 1998. Wavelets that map integers to integers. *Applied Computer Harmonic Analysis*, Vol. 5, No. 3, pp. 332–369.
63. Calderbank, R., I. Daubecheis, W. Sweldens, and B-L.Yeo, 1997. Lossless image compression using integer to integer wavelet transform. In *International Conference on Image Processing*, Vol. I, pp. 596–599.
64. Sheng, F., A. Bilgin, P. Sementilli, and M. Marcellin, 1998. Lossy and lossless image compression using reversible integer wavelet transforms. In *International Conference on Image Processing*, Los Alamitos, CA, Vol. 3, pp. 876–880.
65. Adams, M., and Kossentini, 2000. Reversible integer to integer transforms for image compression: Performance evaluation and analysis. *IEEE Transactions on Image Processing*, Vol. 9, pp. 1010–1024, June 2000.
66. Reichel, J., G. Menegaz, M. Nadenau, and M. Kunt, 2001. Integer wavelet transform for embedded lossy to lossless image compression. *IEEE Transactions on Image Processing*, Vol. 10, pp. 383–392, March 2001.
67. Boulgouris, N., D. Tzovaras, and M. Strintzis, 2001. Lossless image compression based on optimal prediction, adaptive lifting and conditional arithmetic coding. *IEEE Transactions on Image Processing*, Vol. 10, pp. 1–13, January 2001.
68. Cheung, K.-W., C.-H. Cheung, and W.-L. Po, 1999. A novel multiwavelet-based integer transform for lossless image coding. In *International Conference on Image Processing*, Vol. 1, pp. 444–447.
69. Grangetto, M., E. Magli, and G. Olmo, 2000. Minimally non-linear integer wavelets for image coding. In *International Conference on Accoustics, Speech and Signal Processing*, Vol. 4, pp. 2039–2042.

70. Marpe, D., G. Blättermann, J. Ricke, and P. Maaß, 2000. A two-layered wavelet-based algorithm for efficient lossless and lossy image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, pp. 1094–1102, October 2000.
71. Egger, O., and M. Kunt, 1995. Embedded zero tree based lossless image coding. In *International Conference on Image Processing*, Washington, DC, Vol. 3, pp. 616–619.
72. Said, A., and W. Pearlman, 1996. An image multiresolution representation for lossless and lossy compression. *IEEE Transactions on Image Processing*, Vol. 5, pp. 1303–1310, September 1996.
73. Creusere, C., 1998. Successive coefficient refinement for embedded lossless image compression. In *International Conference on Image Processing*, Vol. 1, pp. 521–525.
74. Ramaswamy, V., N. Ranganathan, and K. Namuduri, 1999. Performance analysis of wavelets in emebedded zerotree-based lossless image coding schemes. *IEEE Transactions on Signal Processing*, Vol. 47, pp. 884–889, March 1999.