A Successively Refinable Lossless Image-Coding Algorithm

İsmail Avcibaş, *Member, IEEE*, Nasir Memon, *Member, IEEE*, Bülent Sankur, *Senior Member, IEEE*, and Khalid Sayood

***Abstract*—We present a compression technique that provides progressive transmission as well as lossless and near-lossless compression in a single framework. The proposed technique produces a bit stream that results in a progressive, and ultimately lossless, reconstruction of an image similar to what one can obtain with a reversible wavelet codec. In addition, the proposed scheme provides near-lossless reconstruction with respect to a given bound, after decoding of each layer of the successively refinable bit stream. We formulate the image data-compression problem as one of successively refining the probability density function (pdf) estimate of each pixel. Within this framework, restricting the region of support of the estimated pdf to a fixed size interval then results in near-lossless reconstruction. We address the context-selection problem, as well as pdf-estimation methods based on context data at any pass. Experimental results for both lossless and near-lossless cases indicate that the proposed compression scheme, that innovatively combines lossless, near-lossless, and progressive coding attributes, gives competitive performance in comparison with state-of-the-art compression schemes.**

***Index Terms*—Embedded bit stream, image compression, lossless compression, near-lossless compression, probability mass estimation, successive refinement.**

I. Introduction

LOSSLESS or reversible compression refers to compression techniques in which the reconstructed data exactly matches the original. Near-lossless compression denotes compression methods, which give quantitative bounds on the nature of the loss that is introduced. Such compression techniques provide the guarantee that no pixel difference between the original and the compressed image is above a given value [1]. Both lossless and near-lossless compression find potential applications in remote sensing, medical and space imaging, and multispectral image archiving. In these applications, the volume of the data would call for lossy compression for practical storage or transmission. However, the necessity to preserve the validity and precision of data for subsequent reconnaissance, diagnosis operations, forensic analysis, as well as scientific or clinical measurements, often imposes strict constraints on the reconstruction error. In such situations, near-lossless compression becomes a viable solution, as, on the one hand, it provides significantly higher compression gains vis-à-vis lossless algorithms, and on the other hand, it provides guaranteed bounds on the nature of loss introduced by compression.

Another way to deal with the lossy-lossless dilemma faced in applications such as medical imaging and remote sensing is to use a successively refinable compression technique that provides a bit stream that leads to a progressive reconstruction of the image. Using wavelets, for example, one can obtain an embedded bit stream from which various levels of rate and distortion can be obtained. In fact, with reversible integer wavelets, one gets a progressive reconstruction capability all the way to lossless recovery of the original. Such techniques have been explored for potential use in teleradiology, where a physician typically requests portions of an image at increased quality (including lossless reconstruction) while accepting initial renderings and unimportant portions at lower quality, and thus reducing the overall bandwidth requirements. In fact, the new still-image compression standard, JPEG 2000, provides such features in its extended form [2].

Although reversible integer wavelet-based image-compression techniques provide an integrated scheme for both lossless and lossy compression, the resulting compression performance is typically inferior to the state-of-the-art nonembedded and predictively encoded techniques like CALIC [3] and TMW [4], [5]. Another drawback is that, while lossless compression is achieved when the entire bit stream has been received, for the lossy reconstructions at the intermediate stages, no precise bounds can be set on the extent of distortion present. Near-lossless compression in such a framework is only possible either by an appropriate prequantization of the wavelet coefficients and lossless transmission of the resulting bit stream, or by truncation of the bit stream at an appropriate point followed by transmission of a residual layer to provide the near-lossless bound. Both these approaches have been shown to provide inferior compression, as compared with near-lossless compression in conjunction with predictive coding [1].

In this paper, we present a compression technique that incorporates the above two desirable characteristics, namely, near-lossless compression and progressive refinement from lossy to lossless reconstruction. In other words, the proposed technique produces a bit stream that results in a progressive reconstruction of the image similar to what one can obtain with a reversible wavelet

Paper approved by K. Illgner, the Editor for Speech, Image, Video, and Signal Processing of the IEEE Communications Society. Manuscript received August 7, 2002; revised May 31, 2003 and February 22, 2004. This work was supported in part by the National Science Foundation under INT 9996097. The work of İ. Avcibaş was supported in part by TUBİTAK BDP program. The work of K. Sayood was supported by NASA Goddard Space Flight Center.

İ. Avcibaş is with the Electrical and Electronics Engineering Department, Uludag University, 16059 Bursa, Turkey (e-mail: avcibas@uludag.edu.tr).

N. Memon is with the Computer Science Department, Polytechnic University, Brooklyn, NY 11201 USA (e-mail: memon@poly.edu).

B. Sankur is with the Electrical and Electronics Engineering Department, Bogaziçi University, İstanbul, Turkey (e-mail: sankur@boun.edu.tr).

K. Sayood is with the Electrical Engineering Department, University of Nebraska at Lincoln, Lincoln, NE 68588-0511 USA (e-mail: ksayood@eecomm.unl.edu).

Digital Object Identifier 10.1109/TCOMM.2005.843421

codec. In addition, our scheme provides near-lossless (and lossless) reconstruction with respect to a given bound after each layer of the successively refinable bit stream is decoded. Note, however, that these bounds need to be set at compression time and cannot be changed during decompression. The compression performance provided by the proposed technique is comparable to the best-known lossless and near-lossless techniques proposed in the literature. It should be noted that to the best knowledge of the authors, this is the first technique reported in the literature that provides lossless and near-lossless compression, as well as progressive reconstruction, all in a single framework.

The rest of this paper is organized as follows. We first discuss our approach to near-lossless compression and the tools used in our algorithm, such as successive refinement, density estimation, and the data model in Section II. The proposed compression method is described in Section III. In Section III, we give experimental results, and Section IV concludes the paper.

II. Formulation of Our Approach

The key problem in lossless compression involves estimating the probability density function (pdf) of the current pixel based on previously known pixels (or previously received information). With this in mind, the problem of successive refinement can then be viewed as the process of obtaining improved estimates of the pdf of each pixel with successive passes on the image, until all the pixels are uniquely determined. The fact of restricting the "*support*" (that is the interval where the pdf is nonzero) of the pdf to a successively refined set of intervals leads to the integration of lossless/near-lossless compression in a single framework. More explicitly, diminishing the support of the pdf in each pass to a narrower interval gives progressiveness, while fixing the size of the interval provides near-lossless (or lossless, if the interval size is one) coding. In this unified scheme, we obtain a bit stream that gives us a near-lossless reconstruction after each pass, in the sense that each pixel is within k quantal bins of its original value. The value of this coding-error bound, k, decreases with successive passes, and if desired, in the final pass, we can achieve lossless compression.

In order to design a compression algorithm with the properties described above, we need three things.

1) Given a pixel neighborhood and a gray-level interval, estimate the pdf of the pixel in that interval.
2) Given a pdf for a pixel, how to best predict its actual value or the subinterval in which it is found.
3) Update the pdf of a pixel, given the pdfs of neighborhood pixels.

Equitz and Cover [7] have discussed the problem of successive refinement of information from a rate-distortion point of view. They show that the rate-distortion problem is successively refinable if and only if individual solutions to the rate-distortion problem can be written as a Markov chain. One example process that admits successive refinement is a Gaussian source, together with the mean-square error (MSE) criterion. Hence, for the first requirement in our compression scheme, we adopt the Gaussian data model in the first pass. We assume the data is stationary and Gaussian in a small neighborhood, and therefore, we use linear prediction. We fit a Gaussian density function for the current pixel, with the linear prediction value taken as the optimal estimate of its mean, and the mean-square prediction error as its variance, as detailed in Section II-A. However, in the succeeding passes, we relax the Gaussian assumption.

For the second requirement, we use a technique based on Massey's optimal guessing principle [6]. Let a discrete random variable X be characterized by a set of allowable values $x_1, \ldots, x_n$. Massey [6] has observed that the average number of guesses to determine the value of a discrete random variable is minimized by a strategy that guesses the possible values of the random variable in decreasing order of probability. The guessing game is pursued till the question of the form "Is X equal to x_i?" is positively answered for lossless compression, and the question "Does X lies within δ neighborhood of x_i?" is satisfied for the near-lossless scheme. Therefore, we consider the coding problem as one of asking the optimal sequence of questions to determine the exact value of the pixel, or the interval in which the pixel lies, depending on whether we are interested in lossless or near-lossless compression. Specifically, we divide the support of the current pixel's pdf, initially [0,255], into a nonoverlapping set of intervals $(i, i+\lambda)$, where the search interval is twice the uncertainty interval, that is, $\lambda = 2\delta + 1$. The intervals are then sorted with respect to their probability mass obtained from the estimated density function. Next, the interval with the highest probability mass is identified, and if the pixel is found to lie in this interval, the probability mass outside the interval is zeroed out, and the bit 1 is fed to the entropy coder. Otherwise, bit zero is sent to the encoder, and we repeat the test if the pixel lies in the next highest probability interval. Every time one receives a negative answer, the probability mass of the failing interval is zeroed out and the pdf normalized to a mass of 1. This process is repeated until the right interval is found. Such a pdf with intervals set to zero is called an "*interval-censored pdf*" in the following. At the end of the first pass, the maximum error in the reconstructed image, $\|e\|_\infty$, is δ since the midpoint of the interval is selected as the reconstructed pixel value.

For the third requirement, in the remaining passes of our compression algorithm, we proceed to refine the pdf for each pixel by narrowing the size of the interval in which it is now known to lie. We investigate various pdf refinement schemes as described in Section II-C. The refinement schemes use both *causal* pixels (i.e., pixels that that have occurred before a given pixel in a raster scan order) as well as *noncausal* pixels (i.e., the ones occurring afterwards). Note that in this update scheme, the causal pixels already have a refined pdf, but the noncausal pixels yet do not. We do not want to discard the noncausal pixels, as they may complement, along with the causal pixels, some useful information, like the presence of edges and texture patterns. On the other hand, the information provided by the causal pixels is more accurate, as compared with that of pixels yet to be visited on this pass. This differential in their precision is automatically taken into account when we refine the pdf of the current pixel, as will get clear in Section II-C where we present several pdf-refinement techniques. In any case, with every refinement pass over the image, we continue this estimation and refinement procedure and constrain pixels to narrower and narrower intervals (smaller δ's, hence, smaller values of $\|e\|_\infty$) to the required precision, and if needed, all the way to their exact values.

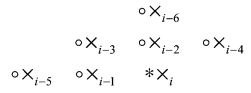


Fig. 1. Ordering of the causal prediction neighbors of the current pixel $\times_i$, $N = 6$.

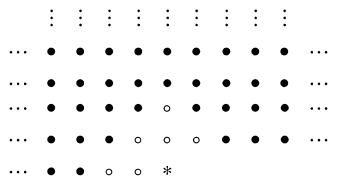


Fig. 2. Context pixels, denoted by $\bullet$ and $\circ$, used in the covariance estimation of the current pixel $*$. The number of context pixels is $K = 40$. These are used in the covariance estimation of the current pixel $*$, as in (3). Each of these 40 context pixels has their causal neighborhoods, as in Fig. 1, for prediction. Some of the causal neighborhood pixels on the context border fall outside the support of the context (these pixels are not shown).

We should note in passing that although image gray values are discrete and their statistics are described with a probability mass function (pmf), we find it convenient to treat them first as continuous random variables. Thus we estimate their pdf, and then revert to their discrete nature during the compression algorithm when we have to compute estimates of their likelihood to lie in a given interval of size.

A. PDF Estimation in the First Pass: The Gaussian Model

Natural images, in general, may not satisfy Gaussianity or stationarity assumptions. But at a coarse level and in a reasonable size neighborhood, the statistics can be assumed to be both Gaussian and stationary, and the Gauss–Markov property can be invoked. Based on this assumption, we fit a Gaussian density for the current pixel. In fact, we use causal neighborhood pixels to predict the current pixel via a linear regression model. Let $p_i(x)$ denote the pdf of the ith pixel at gray value x. Let the probability mass of the mth interval, of length 2δ, and denoted by the upper and lower limits $(L_m, R_m]$, be $p_m = \sum_{x\in(I_m,R_m]} p_i(x)$.

We assume a discrete-time scalar-valued random process $\{X_i\}$. Then $X_{i-1}, X_{i-2}, \ldots, X_{i-N}$ denote the random variables representing the causal neighbors of the current pixel X_i, where the reading order of the pixels is as shown in the mask in Fig. 1. Similarly, let us consider K context pixels of X_i, as shown in Fig. 2, and let us denote them as $X_{i-1}, X_{i-2}, \ldots, X_{i-K}$, in some reading order. For each of these K context pixels, we have to consider their individual causal prediction neighborhood. With this goal in mind, we use the notation with double indexes, x_{i-n}, $i = 1, \ldots, K$; $n = 1, \ldots, N$, where the visited pixel index i runs over the context pixels $(i = 1, \ldots, K)$, and the prediction index n runs over the prediction neighborhood of each context pixel $(n = 1, \ldots, N)$. Thus $\{(x_{(i-1)}, x_{(i-2)}, \ldots, x_{(i-N)}), \ldots, (x_{(i-K-1)}, x_{(i-K-2)}, \ldots x_{(i-K-N)})\}$, $k = 1, \ldots, K$, indicate all the K realizations of the neighborhoods for context pixels. In Fig. 2, the causal mask, shown only for the "star pixel," must be replicated for each of the $K = 40$ other context pixels. Note that each of the K context pixels possesses its own prediction neighborhood consisting of its N causal pixel group, $(x_{k-1}, x_{k-2}, \ldots, x_{k-N})$. This is illustrated in Fig. 2, where each of the $K = 40$ pixels is predicted using a prediction mask as in Fig. 1 (some of the pixels forming the causal neighborhood of the border context pixels are outside the support of the mask shown). Obviously, $N \times K$ pixels are involved for the computation of the regression coefficients. Each of these realizations is assumed to satisfy the Nth-order linear prediction equation. In particular, for the realization $k = 0$, one has

$$X_i = \sum_{j=1}^{N} \beta_j (x_{i-j}) + v_i \tag{1}$$

where $\{\beta_j\}_{j=1}^{N}$ are real-valued linear prediction coefficients of the process, and $\{\nu_i\}$ is a sequence that consists of independent, identically distributed (i.i.d.) random variables having a Gaussian density with zero mean and variance σ^2. Optimal minimum MSE (MMSE) linear prediction for an Nth-order stationary Gauss-Markov process $\{X_i\}$ can be formulated as

$$E[X_i | X_{i-1}, X_{i-2}, \ldots, X_{i-N}] = \sum_{j=1}^{N} \beta_j (x_{i-j}). \tag{2}$$

According to the Gauss–Markov theorem, the minimum variance linear unbiased estimator $\boldsymbol{\beta} = [\beta_1 \ldots \beta_N]$ is the least square solution of the ensemble of equations as (2), resulting from the prediction of the K context pixels, and is given by [8], [9]

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{y}) \tag{3}$$

where $\mathbf{y} = [X_{i-1} X_{i-2}, \ldots, X_{i-K}]$ denote the K context pixels, while the data matrix $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} X_{i-1} & \ldots & X_{i-1-N} \\ \vdots & \ddots & \vdots \\ X_{i-K-1} & \ldots & X_{i-K-N} \end{bmatrix}$$

consists of the prediction neighbors of $X_{i-1}, X_{i-2}, \ldots, X_{i-K}$. The expected value of X_i is given by (2), and an unbiased estimator of prediction error variance σ^2 can be obtained [4] as

$$\sigma^2 = \frac{1}{K - N - 1} \left(\mathbf{y}^T \mathbf{y} - \boldsymbol{\beta}^T \mathbf{X}^T \mathbf{y} \right).$$

Finally, based on the principle that the mean-square prediction for a normal random variable is its mean value, the density of X_i, conditioned on causal neighbors, is then given by

$$p(x_i | x_{i-1}, x_{i-2}, \ldots, x_{i-N}) = \frac{1}{\sqrt{2\pi}\sigma} \times \exp\left(- \left(\frac{1}{2\sigma^2} \right) \left[x_i - \sum_{j=1}^{N} \beta_j (x_{i-j}) \right]^2 \right). \tag{4}$$

B. Encoding the Interval

We can treat the problem of determining the interval where the current pixel value lies within the framework of Massey's guessing principle, as described in Section II. Let $p_1, \ldots, p_M$ denote the M probabilities, where each probability p_m is associated with an interval $(L_m, R_m]$ that has a length of $2\delta + 1$.

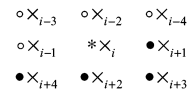


Fig. 3. Causal, $\circ$, and noncausal, $\bullet$, neighbors of the current pixel, $*$, used for probability mass estimation in the second and higher passes.

The number of intervals M depends upon the quantal resolution, and becomes equal to 256 at the finest resolution. The union of these nonoverlapping intervals covers the support of the pdf. We treat the pixel value, whose bin location is being guessed, as a random variable X. The rule for minimizing the number of guesses in finding the interval where the pixel lies is to choose that m for which p_m is maximum. Having thus chosen m as the most likely interval of the pixel, we then use the indicator function $I\{x \in (L_m, R_m]\}$ to test whether the actual pixel value, x, lies in that interval or not. If the mth interval proves to be correct, that is, the pixel is $x \in (L_m, R_m]$, then entropy coder is fed with bit 1, otherwise, if $I\{x \in (L_m, R_m] = 0\}$, the entropy coder is fed bit 0 and the interval with the next highest probability is tested. We used the XYZ adaptive binary arithmetic coder as our entropy coder, and the binary events to be coded were just considered to be an i.i.d. sequence [10]. We would like to note that the arithmetic coder was beneficial only in the initial passes. In the first few passes, the data to be encoded has low entropy, as the strategy of selecting the interval with the highest probability mass proves to be correct most of the time, and the arithmetic coder takes this into account, producing a highly compressed bit stream. The performance improvement in the last pass is marginal, as the data has become more random or less correlated. We expect that context-based arithmetic coding could result in further (but nevertheless small) improvement in the bit rates, as compared with those that we obtained, as reported in Section III.

The average number of guesses that one has to make before one correctly identifies the right interval in which a pixel belongs, gives the number of binary arithmetic coding operations that need to be performed and influences the computational complexity of our technique as well. To give an example, for the Lena image and for the one-pass scheme, one can guess any pixel value in a correct interval of size $\lambda = 7$, within the first three questions with respective cumulative probabilities of 0.73, 0.86, and 0.97. That is, with a 97% chance, we determine the correct gray-level interval of a pixel after the first three guesses.

C. PDF Estimation in the Refinement Passes

After the first pass, we know each pixel value within an uncertainty of $\lambda_1 = 2\delta_1 + 1$. In the successive passes, we proceed to decrease the uncertainty about the value of the pixels. In these finer resolution passes, we drop the Gaussian assumption. Also, as more and more of the redundancy is removed after each pass, this results in decreased correlation in the remaining passes, thereby leading to conditions that cannot be modeled well with a Gaussian assumption. Below, we present three pdf update schemes for the second and higher passes. In these passes, we use both the causal neighborhood densities, which have just been updated, and the noncausal neighborhood densities, which were updated only in the previous pass. The updating neighborhood used in the pdf updates in the second and higher passes can be deduced from Fig. 3.

Method 1: L_2 Norm Minimizing Probability Estimation: This pdf estimate is based on the L_2 norm estimate of the current pixel's pdf, using the pdfs of the causal and noncausal neighborhoods. Let $p_i^*(x)$ denote the pdf to be estimated given the causal and noncausal distributions $p_{i-j}(x), j = -N, \ldots, -1, 1, \ldots N$ (cf. Fig. 3). Notice that in the refinement stage, the prediction neighborhood is different than in the initial stage, as the noncausal pixels also become involved. Minimizing the average difference of $\sum_j (p_i^*(x) - p_{i-j}(x))^2$ subject to the constraint $\int p_i^*(x) = 1$, and using Lagrange multipliers, we have

$$J(p_i^*) = \int \sum_{j=-N}^{N} \left(p_i^*(x) - p_{i-j}(x)\right)^2 dx + \lambda \int p_i^*(x) dx.$$

Using the variational derivative [11] with respect to $p_i^*(x)$, and the information in $p_i(x)$, one finds the updated pdf to be of the form

$$p_i^*(x) = \frac{1}{T}\left(\sum_{j=-N}^{N} p_{i-j}(x)\mathrm{Ind}(p_i(x))\right) \tag{5}$$

where T is the normalizing constant and where $\mathrm{Ind}()$ is an indicator function. In other words, $\mathrm{Ind}(p_i(x)) = 1$ if that quantal level is not censored, and it is zero, $\mathrm{Ind}(p_i(x)) = 0$, if x falls in a censored interval. Thus, the sum in (5) is to be interpreted as an interval-censored averaging of pdfs. Recall that every pixel in the neighborhood after the first pass has an interval-censored pdf. Here we combine the interval-censored pdfs, $p_{i-j}(x)$, defined over one or more intervals of the form $(X_L^i, X_R^i]$, for j in the neighborhood of the ith pixel. If the neighboring interval-censored pdf, say $p_{i-j}(x)$, does not overlap with that of the current one $p_i(x)$, which is being estimated, then it has no contribution on the estimate. In other words, if $p_i(x)$ has an empty bin, it will veto any contribution to the corresponding bin of $p_i^*(x)$ from its neighbors $p_{i-j}(x), j = -N, \ldots, -1, 1, \ldots N$. On the other hand, if that bin was not censored in $p_i(x)$, then evidence accumulation occurs from the corresponding bins of neighbors $p_{i-j}(x), j = -N, \ldots, -1, 1 \ldots, N$. Notice that this method of summing neighboring densities gives implicitly more importance to the relatively more precise pdfs in the causal neighborhoods (1 to N), as they will be concentrated in narrower supports after the latest pass, while the noncausal neighborhoods (-1 to $-N$) will have more dispersed pdfs, that is over a larger support, since their updates are yet to occur.

Method 2: Hellinger Norm Minimizing Probability Estimation: The relative entropy, $D(p \parallel q)$, is a measure of distance between two distributions. In other words, it is a measure of the inefficiency by assuming that the distribution is q when the true distribution is p. For example, if we knew the true distribution of the random variable, then we could construct a code with average length $H(p)$. If instead, we were to use the code for distribution q, we would need $H(p) + D(p \parallel q)$ bits on the average to describe the random variable [12]. The squared Hellinger norm between distributions with densities p and q is defined as

$$H^2(p, q) = \int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx.$$

Many, if not all, smooth function classes satisfy the equivalence $D(p \parallel q) \approx H^2(p, q)$. The advantage of H^2 is that it satisfies the triangle inequality, while D does not [13]. However,

D brings in clean information-theoretic identities, such as minimum description-length principle, stochastic complexity, etc. [13]. Taking advantage of the equivalence between D and H^2, we can use one for the other in the derivation of the optimal $p_i^*(x)$.

When we have a class of candidate densities $\{p_{i-j}(x) : j = 1, \ldots, N\}$ and want to find the $p_i^*(x)$, which minimizes the inefficiency of assuming the distribution was $p_{i-j}(x)$, we can minimize the total extra bits to obtain the shortest description length on the average

$$J(p_i) = \sum_{1 \le j \le N} \int \left(\sqrt{p_i^*(x)} - \sqrt{p_{i-j}(x)} \right)^2 dx + \lambda \int p_i^*(x) dx$$

where λ is the Lagrange multiplier. Again finding the variational derivative with respect to $p_i^*(x)$ and setting it equal to zero, we get

$$p_i^*(x) = \frac{1}{T} \left(\sum_{1 \le j \le N} \sqrt{p_{i-j}(x)} \right)^2 \tag{6}$$

where T is the normalizing constant. In general, the relative entropy or the Kullback–Leibler distance has a close connection with more traditional statistical estimation measures, such as the L_2 norm (MSE) and Hellinger norm, when the distributions are bounded away from zero, and is equivalent to MSE when both p and q are Gaussian distributions with the same covariance structure [13].

The two methods described above were implemented for the pdf update step. Their performances, as measured in terms of the final coding efficiency of the algorithm, were found to be comparable. The proposed L_2 norm pdf update was then adopted due to its computational simplicity.

D. Algorithm Summary

In summary, the proposed algorithm consists of first estimating the pdf of the pixel's grey value, based on the values of the previously decoded pixels, and then to successively refine the estimate by restricting the support of the pdf to narrower and narrower regions in subsequent iterations. The pseudocode of the algorithm is as follows.

```
Encoder
inform the decoder about the number of passes m, near lossless parameters λ_i i = 1,...,m
for each pass
 for each pixel in the image
  if first pass
   predict pdf as per (4)
  else
   predict pdf as per (5)
  end if
  find λ_i length interval with highest probability mass
  while pixel is not in the interval
    entropy code failure event zero
    zero out the probability in the interval
    find λ_i length interval with highest probability mass
  end while
  entropy code success event one
  zero out the probability out of the interval
 end for
end for
```

```
Decoder
Get number of passes m, near lossless parameters λ_i i = 1,...,m
for each pass
 for each pixel in the image
  if first pass
   predict pdf (4)
  else
   predict pdf (5)
  end if
  find λ_i length interval with highest probability mass
  decode the event
  while decoded event is failure
    zero out the probability in the interval
    find λ_i length interval with highest probability mass
    decode the event
  end while
  take midpoint of λ_i length interval as reconstruction value for the pixel
  zero out the probability out of the interval
 end
end
```

III. Experimental Results

In this section, we present simulation results of the performance of our algorithm and compare them with those of its nearest competitors, namely with the CALIC [3], SPIHT [14], and JPEG 2000 [2] algorithms. However, before we do this, we first describe some implementation details and parameter choices used in the specific implementation of our algorithm.

In the implementation of our multipass compression procedure, we first estimate the Gaussian pdf as in (4) not based on the exact pixel values, but on their less precise values resulting from quantization with step size λ_1. Thus, we regress on the quantized version of the context pixels $X_{i-1}, X_{i-2}, \ldots, X_{i-N}$ in (1). In other words, we substitute for each pixel, $\hat{X}_j \leftarrow \text{Midpoint}_{\lambda\_\text{interval}}[X_j]$, the midpoint of the λ-interval within which it is found, where obviously, $\hat{X}_j$ denotes this quantized value. This is necessary, since the same information set must be used at both the encoder and decoder, and the context pixels can be known at the decoder only within precision of δ. In other words, to be consistent, both the encoder and decoder guess the pixel value to be at midpoint of the λ-interval. It can be argued that if one were to use the centroid of each interval instead of the midpoint, the prediction could improve. However, the interval centroid and its midpoint would differ significantly only in the high-sloped intervals of the pdf. In our case, the λ-intervals are small, and furthermore, the correct interval is guessed often in the first few steps, where the relatively flat intervals of the Gaussian pdf around its center is being used. The centroid did not bring in any significant improvement, hence, for the sake of simplicity, we opted for the interval midpoint. We note that, though the initial Gaussian pdf estimate using (4) will be noisier as a consequence of midpoint quantization, the "missed" information will be recovered later using pdf updates based on context pixel pdfs. But the pdf updates necessitate that we pass over the image more than once, hence, the multipass compression. Another consideration is the initialization of the algorithm at the

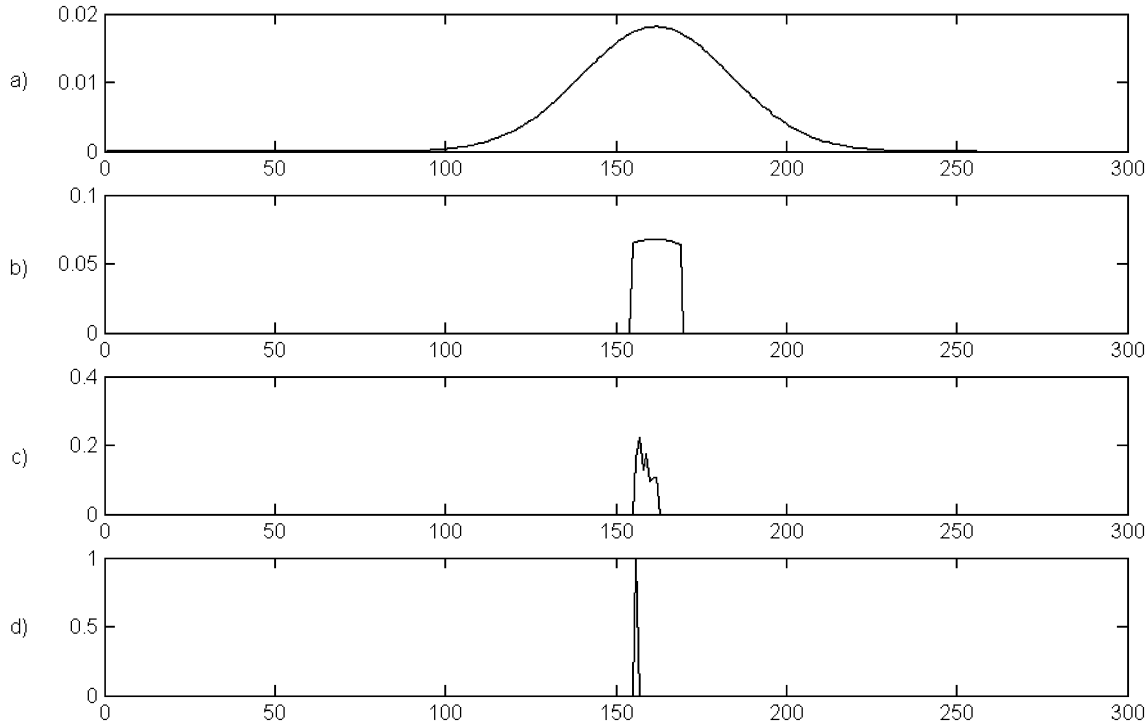


Fig. 4. Example pdf evolution for $\lambda = 15 \rightarrow \lambda = 7 \rightarrow \lambda = 1$ or $\delta = 7 \rightarrow \delta = 3 \rightarrow \delta = 0$). a) After prediction. b) After first pass. c) After second pass. d) After third pass. In three passes, the encoded events are all successes for the given pixel.

borders. When initializing the algorithm, context pixels falling out of the borders of the image are assumed to be zero. Thus initially, the prediction is worse and it costs the algorithm a higher number of transmitted bits, though the overall effect is negligible. The pdf estimation context and the prediction neighborhood we used in the first pass had, respectively, sizes of $K = 40$ and $N = 6$ within the configurations shown in Figs. 1 and 2.

Letting $\lambda_1, \lambda_2, \ldots, \lambda_f \ldots$ denote the sequence of interval lengths used in the progressive coder, and where λ_f is the step size of the final pass, first we seek for the current pixel value within the highest probable λ_1-length interval using the Gaussian pdf estimate based on λ_1-quantized pixels. When the correct interval is determined, the pixel's maximum error becomes δ, since $\lambda_1 = 2\delta$. After we have scanned the whole image, we return back to the second pass to refine the gray-level intervals of the pixels. We thus proceed to narrow down the uncertainty in the gray value of the pixels to a size λ_2 and successively all the way to λ_f, using the minimum norm pdf-update rule as in (5). Recall that in these pdf updates, the Gaussian assumption was abandoned, and addition of the context pixel pdfs was used, censored by the interval of the object pixel. If λ_f is 1, then we obtain lossless coding using multipass procedure; else we obtain near-lossless coding. We report in either case to the arithmetic encoder the failure (0) and success (1) of events (pixel in that interval or not). Fig. 4 shows the evolution of pdf with three passes of the algorithm.

The decoder in our system is similar to the encoder. Based on the estimated pdf and decoded sequences of successes and failures, the pdf support for the current pixel is found. Within the support of the estimated pdf, the intervals are sorted with respect to their probability mass, and the correct interval is found using decoded failure and success events. Recall that the decoded events are the outcomes of the tests whether the current pixel lies in that interval or not. Whenever the decoded event z does not report a success for the given interval, this interval is discarded, and the refining continued until the correct interval is reached. The reconstructed value of the current pixel is taken as the mid-value of the interval reporting success, which guarantees that the error in reconstructed value is not more than half of the interval length minus one, that is, $\|e\|_\infty = (\lambda - 1)/2$, where λ is the length of the interval.

TABLE I
COMPARISON OF THREE LOSSLESS COMPRESSION METHODS FOR $\delta = 0$: BOTH ONE-PASS AND THREE-PASS RESULTS ARE GIVEN WHERE THE INTERVALS ARE TAKEN 8 IN THE FIRST PASS, 4 IN THE SECOND PASS, AND 1 IN THE THIRD PASS. RESULTS ARE IN BITS/PIXEL

	JPEG2000	SPIHT	CALIC	PROPOSED	
$\delta = 0$				3-pass	1-pass
Barbara	4.69	4.58	4.45	4.18	4.21
Lena	4.35	4.20	4.11	4.07	4.07
Zelda	4.02	3.93	3.86	3.81	3.79
Goldhill	4.87	4.78	4.63	4.65	4.69
Boat	4.43	4.34	4.15	4.06	4.08
Mandrill	6.15	5.96	5.88	5.89	5.96
Finger	5.69	5.60	5.60	5.38	5.40
Peppers	4.65	4.58	4.38	4.41	4.37
Average	*4.86*	*4.75*	*4.63*	*4.56*	*4.57*

In Table I, we give the lossless compression performance of our multipass algorithm. The multipass algorithm is arranged in three steps, so that initially bins of size eight are considered, followed by bin size of four, and finally, of size two. For the sake of comparison, we also list results obtained with a single-pass version of our technique, where each pixel value is determined by successive refining down to the precision of one level under the guidance of the Gaussian pdf. We use here the precise values

TABLE II
COMPARISON OF THREE NEAR-LOSSLESS COMPRESSION METHODS ($\delta = 1$)

Image	JPEG2000			SPIHT			CALIC			Proposed		
	Bpp	PSNR	$\|e\|_\infty$	Bpp	PSNR	$\|e\|_\infty$	Bpp	PSNR	$\|e\|_\infty$	bpp	PSNR	$\|e\|_\infty$
Barbara	2.65	47.11	5	2.65	46.90	6	2.94	49.91	1	2.65	49.88	1
Lena	2.58	47.82	5	2.58	48.26	4	2.60	49.90	1	2.58	49.89	1
Zelda	2.25	47.99	5	2.25	48.24	4	2.35	49.90	1	2.25	49.89	1
Goldhill	3.11	47.69	5	3.11	47.81	4	3.09	49.90	1	3.11	49.90	1
Boat	2.63	47.75	6	2.63	47.86	5	2.65	49.89	1	2.63	49.89	1
Mandrill	4.31	47.43	4	4.31	47.65	5	4.31	49.90	1	4.31	49.89	1
Finger	3.80	47.54	5	3.80	47.19	5	3.90	49.89	1	3.80	49.89	1
Peppers	2.90	47.63	5	2.90	48.10	4	2.84	49.90	1	2.90	49.89	1
Average	*3.03*	*47.62*	*5.0*	*3.03*	*47.75*	*4.6*	*3.09*	*49.90*	*1*	*3.03*	*49.89*	*1*

TABLE III
COMPARISON OF THREE NEAR-LOSSLESS COMPRESSION METHODS ($\delta = 3$)

Image	JPEG2000			SPIHT			CALIC			Proposed		
	Bpp	PSNR	$\|e\|_\infty$	Bpp	PSNR	$\|e\|_\infty$	Bpp	PSNR	$\|e\|_\infty$	bpp	PSNR	$\|e\|_\infty$
Barbara	1.61	41.95	12	1.61	41.57	12	1.92	42.23	3	1.61	42.27	3
Lena	1.50	42.66	13	1.50	42.92	8	1.57	42.26	3	1.50	42.29	3
Zelda	1.27	43.31	8	1.27	43.40	8	1.37	42.19	3	1.27	42.25	3
Goldhill	1.98	41.77	12	1.98	41.91	10	2.01	42.19	3	1.98	42.19	3
Boat	1.60	42.91	11	1.60	42.74	9	1.67	42.37	3	1.60	42.43	3
Mandrill	3.11	40.62	13	3.11	41.13	11	3.13	42.10	3	3.11	42.11	3
Finger	2.63	41.03	11	2.63	40.49	15	2.73	42.11	3	2.63	42.10	3
Peppers	1.78	41.96	11	1.78	42.37	8	1.78	42.21	3	1.78	42.21	3
Average	*1.94*	*42.03*	*11.4*	*1.94*	*42.07*	*10.1*	*2.02*	*42.21*	*3*	*1.94*	*42.23*	*3*

TABLE IV
COMPARISON OF THREE NEAR-LOSSLESS COMPRESSION METHODS ($\delta = 7$)

Image	JPEG2000			SPIHT			CALIC			Proposed		
	Bpp	PSNR	$\|e\|_\infty$	Bpp	PSNR	$\|e\|_\infty$	Bpp	PSNR	$\|e\|_\infty$	bpp	PSNR	$\|e\|_\infty$
Barbara	0.91	37.33	22	0.91	36.64	23	1.19	36.21	7	0.91	36.27	7
Lena	0.73	38.81	23	0.73	38.92	16	0.83	36.53	7	0.73	36.68	7
Zelda	0.54	39.81	19	0.54	39.90	15	0.66	36.50	7	0.54	36.80	7
Goldhill	1.15	37.44	18	1.15	37.46	16	1.18	35.86	7	1.15	35.87	7
Boat	0.91	38.54	23	0.91	38.40	19	1.00	36.51	7	0.91	36.50	7
Mandrill	2.09	35.23	25	2.09	35.53	22	2.15	35.50	7	2.09	35.48	7
Finger	1.73	36.29	20	1.73	35.64	21	1.80	35.43	7	1.73	35.43	7
Peppers	0.93	38.04	15	0.93	38.23	15	0.95	36.25	7	0.93	36.27	7
Average	*1.12*	*37.67*	*20.6*	*1.12*	*37.59*	*18.4*	*1.22*	*36.10*	*7*	*1.12*	*36.16*	*7*

of the causal context pixels in deriving the conditional pdf of a pixel given its context. Of course, a one-pass scheme will not be progressive in nature, as each pixel must be refined to the desired δ-accuracy before proceeding to the next one. Nevertheless, it provides a baseline comparison of our approach to other lossless but nonprogressive compression. One can notice that the one-pass and the three-pass modalities yield almost identical results. This is somewhat surprising, in that one would expect the multipass algorithm to yield better scores, since, after the first pass, the noncausal information becomes available. This could be either because the noncausal information is less precise than the causal information, hence, not sufficiently useful and/or the pdf refinement is not as effective.

Near-lossless compression performance results, namely bit rate, peak signal-to-noise ratio (PSNR), and maximum error magnitude $\|e\|_\infty$ are listed in Tables II–IV for the three coders obtained for the near-lossless parameter $\delta = 1$, $\delta = 3$, and $\delta = 7$. (Recall that the interval corresponding to a quantal uncertainty is given by $\lambda = 2\delta + 1$). Results listed are after a single pass of the proposed algorithm. The reason for doing this is to show how the proposed algorithm performs as compared with CALIC, SPIHT, and JPEG 2000 in the intermediate steps of reconstruction. This first pass can then be followed by lossless refinement, and would give results similar to what is shown in Table I. In any case, near-lossless results indicate that the proposed coder has better rate-distortion performance than SPIHT and JPEG 2000 for bounds $\delta = 1$, $\delta = 3$ in PSNR, and $\|e\|_\infty$ criteria. As δ increases to seven, SPIHT and JPEG 2000 starts to catch up with the proposed coder in PSNR, but at the same time, its $\|e\|_\infty$ score is two to three times larger than that of our algorithm or that of CALIC. Such large quantization errors may be objectionable for many applications. The rate-distortion performance vis-à-vis CALIC improves as the bound gets larger, as can be seen from Tables II–IV.

IV. Conclusion

In this paper, we have presented a technique that unifies progressive transmission and near-lossless compression in one single bit stream. The proposed technique produces a bit stream that results in the progressive reconstruction of the image, just like what one can obtain with a reversible wavelet codec. In addition, our scheme provides near-lossless reconstruction with respect to a given bound after each layer of the successively refinable bit stream is decoded. Furthermore, the compression performance provided by the proposed technique is comparable to the best-known lossless and near-lossless techniques proposed in the literature [2], [3], [14].

The originality of the method consists of looking at the image-data compression as one of successively refining the pdf estimate of image pixels. With successive passes of the image, the support of the pdf estimate is gradually narrowed until it becomes of length one, in which case, we have lossless reconstruction. Stopping the process at any intermediate stage gives near-lossless compression. The proposed technique provides a flexible framework, and many variations of the basic method are possible. For example, the quality of reconstruction as defined by the near-lossless parameter λ can be made to vary from region to region, or even from pixel to pixel, based on image content or other requirements. Given this fact, different regions in the image can be refined to different desired precision using human vision system properties. To this effect, flat regions where the visibility of compression artifact is higher can be refined more accurately, thus achieving perceptually near-lossless compression. Variations on the themes of context prediction and pdf update can be envisioned. Work is currently progressing to extend this technique to multispectral images.

References

[1] R. Ansari, N. Memon, and E. Ceran, "Near-lossless image compression techniques," *J. Electron. Imaging*, vol. 7, pp. 486–494, Jul. 1998.

[2] D. Taubman and M. W. Marcellin, *JPEG 2000, Image Compression, Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.

[3] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, pp. 437–444, Apr. 1997.

[4] B. Meyer and P. Tischer, "TMW-A new method for lossless image compression," in *Proc. Picture Coding Symp.*, Berlin, Germany, Oct. 1997, pp. 533–538.

[5] ——, "Extending TMW for near-lossless compression of grayscale images," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 458–470.

[6] J. L. Massey, "Guessing and entropy," in *Proc. 1994 IEEE Int. Symp. Inf. Theory*, Trondheim, Norway, 1994, pp. 204–204.

[7] H. R. Equitz and T. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, pp. 269–275, Mar. 1991.

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

[9] A. C. Rencher, *Methods of Multivariate Analysis*. New York: Wiley, 1995.

[10] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, pp. 520–540, Jun. 1987.

[11] G. B. Thomas, *Calculus and Analytic Geometry*. Reading, MA: Addison-Wesley, 1972.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

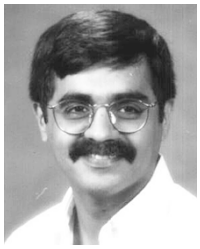
[13] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2743–2760, Oct. 1998.

[14] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–249, Mar. 1996.

İsmail Avcibaş (M'02) received the B.S. and M.S. degrees in electronics engineering from Uludag University, Bursa, Turkey, in 1992 and 1994, respectively, and the Ph.D. degree in electrical and electronics engineering from Bogaziçi University, İstanbul, Turkey, in 2001.

He is currently an Assistant Professor with the department of electronics engineering, Uludag University, Bursa, Turkey. His current research interests are in signal processing, data compression, steganalysis and multimedia communications. He performed research on image compression and steganalysis in the Department of Computer and Information Science, Polytechnic University, Brooklyn, NY, in 1999–2000.

Dr. Avcibaş received a scholarship from The Scientific Council of Turkey TUBITAK, BDP Program.



Nasir Memon (S'92–M'92) is a Professor with the Computer Science Department, Polytechnic University, Brooklyn, NY. His research interests include data compression, computer and network security, and multimedia communication, computing, and security. He is currently an Associate Editor for the *ACM Multimedia Systems Journal* and the *Journal of Electronic Imaging*.

He was an Associate Editor for the IEEE Transactions on Image Processing from 1999–2002.

Bülent Sankur (S'70–M'76–SM'90) received the B.Sc. degree from Bogaziçi University, İstanbul, Turkey, and the M.Sc. and Ph.D. degrees from Rensselaer Polytechnic Institute, Troy, NY.

He has been active in the Department of Electric and Electronic Engineering, Bogaziçi University, establishing curricula and laboratories and guiding research in the areas of digital signal processing, image and video compression, biometry, and multimedia systems. He has held visiting positions at the University of Ottawa, Ottawa, ON, Canada, İstanbul Technical University, the Technical University of Delft, Delft, The Netherlands, and the Ecole Nationale Supérieure des Telecommunications, France.

Dr. Sankur was the Chairman of the 1996 International Telecommunications Conference and the technical Co-Chairman of ICASSP 2000.

Khalid Sayood received the B.S. and M.S. degrees in electrical engineering from the University of Rochester, Rochester, NY, in 1977 and 1979, respectively, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 1982.

He joined the University of Nebraska, Lincoln, in 1982, where he currently serves as the Henson Professor of Engineering. From 1995 to 1996, he served as the Founding Head of the Computer Vision and Image Processing Group at the Turkish National Research Council Informatics Institute (TUBITAK-MAM), and spent the 1996-1997 academic year as Visiting Professor at Bogaziçi University, İstanbul, Turkey. He is the author of *Introduction to Data Compression* (San Mateo, CA: Morgan Kaufmann, 2nd edition) and the Editor of the *Handbook of Lossless Compression* (New York: Academic, to be published). His current research includes joint source/channel coding, biological sequence analysis, and data compression.