Lossless Compression of Continuous-Tone Images

BRUNO CARPENTIERI, MEMBER, IEEE, MARCELO J. WEINBERGER, SENIOR MEMBER, IEEE, AND GADIEL SEROUSSI, FELLOW, IEEE

In this paper, we survey some of the recent advances in lossless compression of continuous-tone images. The modeling paradigms underlying the state-of-the-art algorithms, and the principles guiding their design, are discussed in a unified manner. The algorithms are described and experimentally compared.

Keywords*—*Adaptive coding, compression standards, data compression, image coding, lossless compression.

I. INTRODUCTION

Compression is the coding of data to minimize its representation. The compression of images is motivated by the economic and logistic needs to conserve space in storage media and to save bandwidth in communication. The compression process is termed *lossless* (also reversible or noiseless) if the reconstructed image is identical to the original; otherwise, it is called *lossy* compression (also irreversible or noisy). One can expect to be able to losslessly compress images since the data contains redundancies, in the sense that it can be efficiently modeled using nonuniform distributions.

Lossless image compression is required (or desired) in applications where the pictures are subject to further processing (e.g., for the purpose of extraction of specific information), intensive editing, or repeated compression/decompression. It is generally the choice also for images obtained at great cost, or in applications where the desired quality of the rendered image is yet unknown. Thus, medical imaging, prepress industry, image archival systems, precious artworks to be preserved, and remotely sensed images are all candidates for lossless compression.

Images are defined in [16] simply as a set of two-dimensional arrays of integer data (the *samples*), represented with a given precision (number of bits per component). Each array is termed a *component*, and color images have multiple components, which usually result from a representation in some color space (e.g., RGB, YUV, CMYK). A continuous-tone image, in turn, is an image whose components have more than one bit per sample. However, this wide definition is not meant to imply that the JPEG-LS standard specified in [16] (or any other state-of-the-art lossless compression algorithm for continuous-tone images) was designed to handle any such array. As discussed later in this section, most successful schemes build on certain assumptions about the image data, and may perform poorly in case these assumptions do not hold. For example, palletized images (which have a single component, representing an array of indices to a palette table, rather than multiple components as in the original color space representation) qualify as continuous-tone according to the above definition. However, the compression results using JPEG-LS might be poor unless the palette table is appropriately arranged prior to compression.

While many modern applications of lossless image compression deal mainly with color images, our discussions will generally be confined to a single component. The tools employed in the compression of color images are derived from those developed for each component, and their integration is discussed in Section VII.

State-of-the-art lossless image compression schemes can be subdivided into two distinct and independent phases, *modeling* and *coding*, which are often interleaved. An image is observed sample by sample in some predefined order (normally raster-scan), and the modeling phase is aimed at gathering information about the data, in the form of a probabilistic model, which is used for coding. Modeling can be formulated as an inductive inference problem, in which at each time instant t, and after having scanned past data $x^t = x_1 x_2 \cdots x_t$, one wishes to make inferences on the next sample value x_{t+1} by assigning a conditional probability distribution $P(\cdot|x^t)$ to it. Ideally, the code length contributed by x_{t+1} is $-\log P(x_{t+1}|x^t)$ bits (logarithms are taken to the base 2), which averages to the *entropy* of the probabilistic model. This code assignment is implemented in the coding phase. In a *sequential* (or on-line) scheme, the

Manuscript received February 15, 2000; revised June 26, 2000.

B. Carpentieri is with the Dipartimento di Informatica ed Applicazioni "R. M. Capocelli," Universitá di Salerno, 84081 Baronissi (SA), Italy (e-mail: bc@dia.unisa.it).

M. J. Weinberger and G. Seroussi are with the Information Theory Research Group, Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: marcelo@hpl.hp.com; seroussi@hpl.hp.com).

Publisher Item Identifier S 0018-9219(00)09993-X.

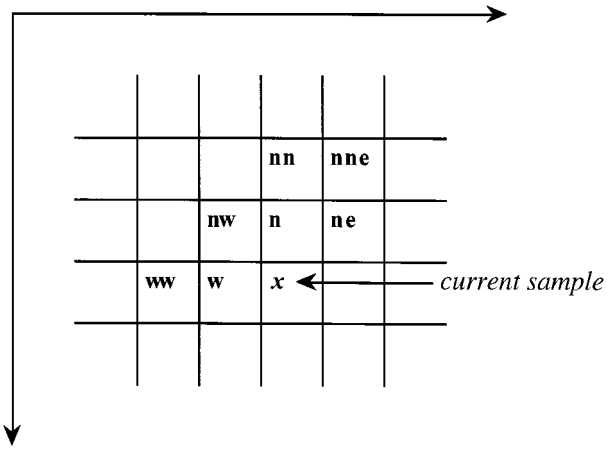

Fig. 1. Labeling of neighboring samples in a causal template.

model for encoding x_{t+1} is *adaptive* and uses the values of the previously coded samples x^t. Thus, it can be mimicked by the decoder as it decodes the past string, sequentially. Alternatively, in a *two-pass* scheme, a model is learned from the whole image in a first pass, and must be described to the decoder as header information.

The conceptual separation between the modeling and coding operations [40] was made possible by the advent of the *arithmetic codes* [37] (see, e.g., [59] for an overview of arithmetic coding), which can realize any model $P(\cdot|\cdot)$ to a preset precision. These two milestones in the development of lossless data compression allowed researchers to view image coding merely as a probability assignment problem, concentrating on the design of imaginative models for images. However, when the complexity axis is considered, the above separation between modeling and coding becomes less clean. This is because the use of a generic arithmetic coder, which enables the most general models, is ruled out in many low-complexity applications, especially for software implementations.

Modern lossless image compression, which is based on the above modeling/coding paradigm, can be traced back to the Sunset algorithm [50] in the early 1980s. Moreover, most state-of-the-art schemes are still based on a modeling strategy pioneered by Sunset, in which the adaptive probability assignment is broken into the following components.

1) A prediction step, in which a value $\hat{x}_{t+1}$ is guessed for the next sample x_{t+1} based on a finite subset (a *causal template*) of the available past data x^t.
2) The determination of a *context* in which x_{t+1} occurs. The context is, again, a function of a (possibly different) causal template.
3) A probabilistic model for the *prediction residual* (or error signal) $\epsilon_{t+1} \stackrel{\Delta}{=} x_{t+1} - \hat{x}_{t+1}$, conditioned on the context of x_{t+1}.

The prediction residual is coded based on the probability distribution designed in Step 3) above. The same prediction, context, and probability distribution are available to the decoder, which can then recover the exact value of the encoded sample. A causal template is shown in Fig. 1, where locations are labeled according to their position with respect to the current sample. Henceforth, the sample intensity value at a generic location a is denoted I_a.

The prediction step is, in fact, one of the oldest and most successful tools in the image compression toolbox. It is particularly useful in situations where the samples represent a continuously varying physical magnitude (e.g., brightness), and the value of the next sample can be accurately predicted using a simple function (e.g., a linear combination) of previously observed neighboring samples. The usual interpretation of the beneficial effect of prediction is that it *decorrelates* the data samples, thus, allowing the use of simple models for the prediction errors.

As for the context determination, in the spirit of Occam's Razor, the number of parameters K in the statistical model plays a key role that was already understood in Sunset. The designer aims at a code length that approaches the empirical entropy of the data under the model. Lower entropies can be achieved through larger contexts (namely, a larger value of K), by capturing high-order dependencies, but the entropy savings could be offset by a *model cost*. This cost captures the penalties of "context dilution" occurring when count statistics must be spread over too many contexts, thus, affecting the accuracy of the corresponding estimates. For parametric models, and under mild regularity conditions, the per-sample asymptotic model cost was quantified in [38] as $(K \log n)/(2n)$, where n is the number of data samples. In two-pass schemes, the model cost represents the code length required to describe the model parameters estimated in the first pass, which must be transmitted to the decoder. In a context model, K is determined by the number of free parameters defining the coding distribution at each context and by the number of contexts.

In order to balance the above "tension" between entropy savings and model cost, the choice of model should be guided by the use, whenever possible, of available *prior knowledge* on the data to be modeled, thus, avoiding unnecessary "learning" costs (i.e., overfitting). This explains the relative failure of *universal* compression tools based on the Lempel–Ziv (LZ) algorithm [68], [69] when applied directly to natural images, and the need for schemes specifically designed for image data. For example, a fixed linear predictor reflects prior knowledge on the smoothness of the data. Prior knowledge can be further utilized by fitting parametric distributions with few parameters per context to the data. This approach allows for a larger number of contexts to capture higher order dependencies without penalty in overall model cost.

Successful modern lossless image coders, such as the new JPEG-LS standard [56] and context-based, adaptive, lossless image coder (CALIC) [65], make sophisticated assumptions on the data to be encountered. These assumptions are not too strong, though, and while these algorithms are primarily targeted at photographic images, they perform very well on other image types such as compound documents, which may also include text and graphic portions. However, given the great variety of possible image sources, it cannot be expected from a single algorithm to optimally handle all types of images. Thus, there is a tradeoff between scope and

compression efficiency for specific image types, in addition to the tradeoffs between compression efficiency, computational complexity, and memory requirements. Consequently, the field of lossless image compression is still open to fruitful research.

In this paper, we survey some of the recent advances in lossless compression of continuous-tone images. After reviewing pioneering work in Sections II and III, Sections IV and V present state-of-the-art algorithms as the result of underlying modeling paradigms that evolved in two different directions. One approach emphasizes the compression aspect, aiming at the best possible compression ratios. In the other approach, judicious modeling is used to show that state-of-the-art compression is not incompatible with low complexity. Other interesting schemes are summarized in Section VI. Section VII discusses the compression of color and palletized images. Finally, Section VIII presents experimental results.

II. The Sunset Algorithm and Lossless JPEG

A variant of the Sunset algorithm was standardized in the first lossless continuous-tone image compression standardization initiative. In 1986, a collaboration of three major international standards organizations (ISO, CCITT, and IEC), led to the creation of a committee known as Joint Photographic Experts Group (JPEG), with a charter to develop an international standard for compression and decompression of continuous-tone, still frame, monochrome, and color images. The resulting JPEG standard includes four basic compression methods: three of them (sequential encoding, progressive encoding, hierarchical encoding) are based on the discrete cosine transform (DCT) and used for lossy compression, and the fourth, lossless JPEG [52], [35], [20], has one mode that is based on Sunset.

In Lossless JPEG, two different schemes are specified: one using arithmetic coding, and one for Huffman coding. Seven possible predictors combine the values of up to three neighboring samples, (I_w, I_{nw}, I_n), to predict the current sample: I_w, I_n, I_{nw}, $I_w + I_n - I_{nw}$, $I_w + ((I_n - I_{nw})/2)$, $I_n + ((I_w - I_{nw})/2)$, and $(I_w + I_n)/2$. The predictor is specified "up front," and remains constant throughout the image.

The arithmetic coding version, a variant of Sunset, groups prediction errors into five categories and selects the context based on the category for the prediction errors previously incurred at neighboring locations w and n, for a total of 25 contexts. The number of parameters is further reduced by fitting a probabilistic model with only four parameters per context, to encode a value $x \in \{0, 1, -1\}$, or one of two "escape" codes if $|x| > 1$. Values $|x| > 1$ are conditioned on only two contexts, determined by the prediction error at location n, and are grouped, for parameter reduction, into "error buckets" of size exponential in $|x|$. The total number of parameters for 8-bit/sample images is 128. Each parameter corresponds to a *binary* decision, which is encoded with an adaptive binary arithmetic coder.

The Huffman version is a simple DPCM scheme, in which errors are similarly placed into "buckets" that are encoded with a Huffman code, using either recommended (fixed) or customized tables. The index of the prediction error within the bucket is fixed-length code. Given the absence of a context model, these simple techniques are fundamentally limited in their compression performance by the first-order entropy of the prediction residuals, which, in general, cannot achieve total decorrelation of the data [9]. Even though the compression gap between the Huffman and the arithmetic coding versions is significant, the latter did not achieve widespread use, probably due to its higher complexity requirements and to intellectual property issues concerning the use of the arithmetic coder.

At the time of this writing, the Independent JPEG Group's lossless JPEG image compression package, a free implementation of the lossless JPEG standard, is available by anonymous ftp from ftp://ftp.cs.cornell.edu/pub/multimed/ljpg.

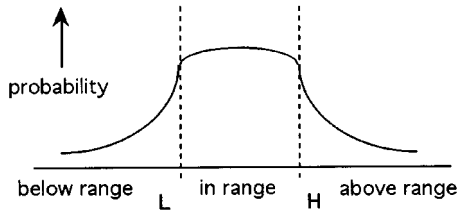

Fig. 2. Probability distribution assumed by FELICS.

III. FELICS: Toward Adaptive Models at Low Complexity

The algorithm FELICS (fast, efficient lossless image compression system) [14], [15] can be considered a first step in bridging the compression gap between simplicity-driven schemes, and schemes based on context modeling and arithmetic coding, as it incorporates adaptivity in a low-complexity framework. Its compression on photographic images is not far from that provided by lossless JPEG (arithmetic coding version), and the code is reported in [14] to run up to five times faster. For complexity reasons, FELICS avoids the use of a generic arithmetic code, and uses at least one bit per each coded sample. Consequently, it does not perform well on highly compressible images.

The key idea is to assume, for each sample, a probability distribution as depicted in Fig. 2, where L and H denote, respectively, the minimum and maximum between the sample values at locations w and n, and the decay on both sides is assumed exponential and symmetric. Thus, FELICS departs from the traditional predictive coding paradigm in that not one, but two "prediction values" are used. The rate of decay depends on a context, determined by the difference $\Delta = H - L$. A probability close to 0.5 is assumed for the central part of the distribution, leading to a coding scheme that uses one bit to indicate whether the value x of the current sample lies between H and L. In this case, an adjusted binary code is used, as the distribution is assumed nearly flat. Otherwise, an additional bit indicates whether x is above or below the interval $[L, H]$, and a Golomb code specifies the value of $x - H - 1$ or $L - x - 1$. Golomb codes are the optimal

(Huffman) codes for geometric distributions of the nonnegative integers [10], and were first described in [12] as a means for encoding run lengths. Given a positive integer parameter m, the mth order Golomb code encodes an integer $y \geq 0$ in two parts: a *unary* representation of $\lfloor y/m \rfloor$, and a *modified binary* representation of $y \bmod m$ (using $\lfloor \log m \rfloor$ bits if $y < 2^{\lceil \log m \rceil} - m$ and $\lceil \log m \rceil$ bits otherwise). FELICS uses the special case of Golomb codes with $m = 2^k$, which leads to very simple encoding/decoding procedures, as already noted in [12]. The parameter k is chosen as the one that would have performed best for the previous occurrences of the context.

At the time of this writing, the mg-FELICS freeware implementation can be obtained as part of the mg package (see [58]) from http://www.cs.mu.oz.au/mg/

IV. Modeling for High-Performance Compression with Arithmetic Coding

The optimization of the modeling steps outlined in Section I, inspired on the ideas of *universal modeling*, is analyzed in [53], where a concrete scheme (albeit of relatively high complexity) is presented as a way to demonstrate these ideas. Following [26], we will refer to this scheme as UCM (for universal context modeling). Image compression models, such as Sunset, customarily consisted of a fixed structure, for which parameter values were adaptively learned. The model in [53], instead, is adaptive not only in terms of the parameter values, but also in their number and structure. In UCM, the context for x_{t+1} (a sample indexed using raster scan order) is determined out of differences $x_{t_i} - x_{t_j}$, where the pairs (t_i, t_j) correspond to adjacent locations within a fixed causal template, with $t_i, t_j \leq t$. Each difference is adaptively quantized based on the concept of stochastic complexity [39], to achieve an optimal number of contexts. The prediction step is accomplished with an adaptively optimized, context-dependent function of neighboring sample values. In addition to the specific linear predicting function demonstrated in [53] as an example of the general setting, implementations of the algorithm include an affine term, which is estimated through the average of past prediction errors, as suggested in [53, p. 583].

To reduce the number of parameters, the model for arithmetic coding follows the widely accepted observation that prediction residuals ϵ in continuous-tone images, conditioned on the context, are well modeled by a *two-sided geometric distribution* (TSGD). For this two-parameter class, the probability decays exponentially according to $\theta^{|\epsilon+\mu|}$, where $\theta \in (0, 1)$ controls the two-sided decay rate and μ reflects a DC offset typically present in the prediction error signal of context-based schemes.[1] The resulting code length for the UCM model is asymptotically optimal in a certain broad class of processes used to model the data.

While UCM provided the best published compression results at the time (at the cost of high complexity), it could be argued that the improvement over the fixed model structure paradigm, best represented by the Sunset family of algorithms (further developed in [21] and [22]), was scant. However, the image modeling principles outlined in [53] shed light on the workings of some of the leading lossless image compression algorithms. In particular, the use of adaptive, context-based prediction, and the study of its role, lead to a basic paradigm proposed and analyzed in [70], [54], in which a multiplicity of predicting contexts is clustered into a smaller set of conditioning states. This paradigm is summarized next.

While Sunset uses a fixed predictor, in general, a predictor consists of a fixed and an adaptive component. When the predictor is followed by a zeroth-order coder (i.e., no further context modeling is performed, as in the Huffman version of lossless JPEG), its contribution stems from it being the only "decorrelation" tool in the compression scheme. When used in conjunction with a context model, however, the contribution of the predictor is more subtle, especially for its adaptive component. In fact, prediction may seem redundant at first, since the same contextual information that is used to predict is also available for building the coding model, which will eventually learn the "predictable" patterns of the data and assign probabilities accordingly. The use of two different modeling tools based on the same contextual information is analyzed in [70], [54], and the interaction is explained in terms of model cost. The first observation is that prediction turns out to reduce the number of coding parameters needed for modeling high-order dependencies. This is due to the existence of multiple conditional distributions that are similarly shaped but centered at different values. By predicting a deterministic, context-dependent value $\hat{x}_{t+1}$ for x_{t+1}, and considering the (context)-conditional probability distribution of the prediction residual ϵ_{t+1} rather than that of x_{t+1} itself, we allow for similar probability distributions on ϵ, which may now be all centered at zero, to merge in situations when the original distributions on x would not. Now, while the fixed component of the predictor is easily explained as reflecting our prior knowledge of typical structures in the data, leading, again, to model cost savings, the main contribution in [70], [54] is to analyze the adaptive component. Notice that adaptive prediction also learns patterns through a model (with a number K' of parameters), which has an associated learning cost. This cost should be weighted against the potential savings of $O(K(\log n)/n)$ in the coding model cost. A first indication that this tradeoff might be favorable is given by the bound in [28], which shows that the per-sample model cost for prediction is $O(K'/n)$, which is asymptotically negligible with respect to the coding model cost discussed in Section I. The results in [70], [54] confirm this intuition and show that it is worth increasing K' while reducing K. As a result, [70], [54] proposes the basic paradigm of using a large model for adaptive prediction, which, in turn, allows for a smaller model for adaptive coding. This paradigm is also applied, for instance, in CALIC [65].

CALIC [60], [65], [61] avoids some of the optimizations performed in UCM, but by tuning the model more carefully to the image compression application, some compression

[1] In fact, UCM assigns to a sample x a probability resulting from integrating a Laplacian distribution in the interval $[x - 0.5, x + 0.5)$. This probability assignment departs slightly from a TSGD.

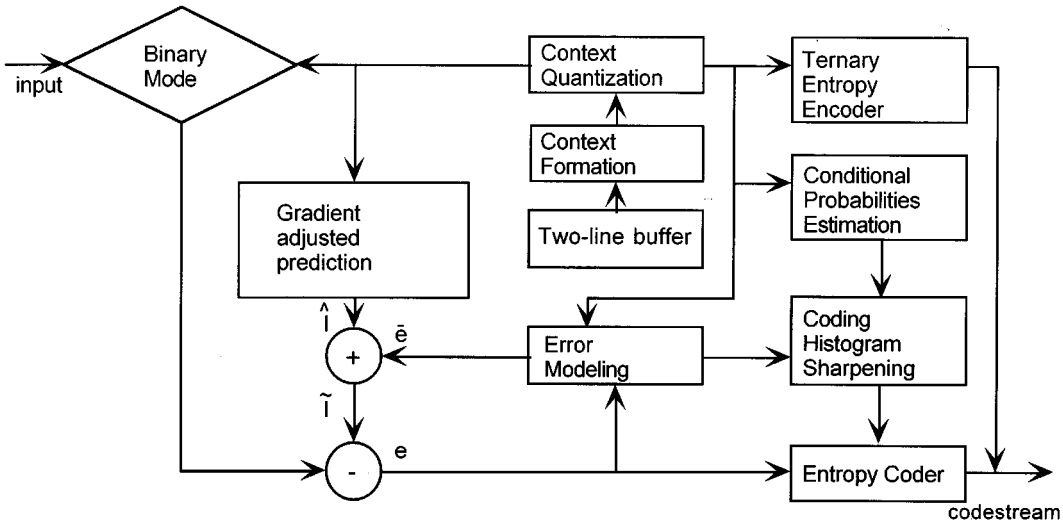

Fig. 3. CALIC's encoder.

gains are obtained. A block diagram of the encoder is shown in Fig. 3 (from [65]). CALIC has two modes of operation: a binary mode that is used when an already encoded neighborhood of the sample to be coded has no more than two distinct intensity values, and a continuous-tone mode.

In binary mode, a ternary event (including an escape symbol that causes a switchback to continuous-tone mode) is coded using a context-based arithmetic coder. In the continuous-tone mode, the prediction step is context-dependent, but differs from [53] in that it incorporates prior knowledge through a fixed component, termed GAP (gradient adjusted predictor), which switches between combinations of neighboring sample values based on local gradient information. The context-dependent adaptive correction is similar to the affine term in classical autoregressive (AR) models. Let d_h and d_v denote estimates, within a scaling factor, of the gradient magnitude near the current location in the horizontal and vertical directions, respectively, given by

$$d_h = |I_w - I_{ww}| + |I_n - I_{nw}| + |I_n - I_{ne}|$$
$$d_v = |I_w - I_{nw}| + |I_n - I_{nn}| + |I_{ne} - I_{nne}|.$$

These estimates detect the orientation and magnitude of edges, and determine the weights assigned to neighboring sample values in the calculation of the GAP prediction $\hat{x}$, as follows:

```
if (d_v − d_h > 80) x̂ = I_w /* sharp horizontal
  edge */
else if (d_v − d_h < −80) x̂ = I_n /* sharp ver-
  tical edge */
else {
  x̂ = (I_w + I_n)/2 + (I_ne − I_nw)/4;
  if (d_v − d_h > 32) x̂ = (x̂+I_w)/2 /* horizontal
  edge */
  else /* weak horizontal edge */
    if (d_v − d_h > 8) x̂ = (3x̂ + I_w)/4
  else /* vertical edge */
    if (d_v − d_h < −32) x̂ = (x̂ + I_n)/2
  else /* weak vertical edge */
    if (d_v − d_h < −8) x̂ = (3x̂ + I_n)/4;
}
```

The context for the adaptive part of the predictor has components for "local texture" and "error energy." The texture component is based on the sign of the differences between the value of neighboring samples and the GAP prediction. The energy component is computed by quantizing a linear combination of d_v, d_h, and previous GAP prediction errors. These two components are combined in a Cartesian product to form the compound modeling contexts. The GAP prediction is then corrected with an adaptive term, which is estimated through the average of past prediction errors at the context. Overall, CALIC uses the seven surrounding pixels shown in Fig. 1 for prediction and context modeling.

CALIC adheres to the paradigm of using a large collection of contexts for adaptive prediction, and few conditioning contexts for coding, by restricting the latter to the error energy component. In total, 576 and eight contexts, respectively, are used for 8-bit/sample images. Since conditional prediction error distributions from different contexts merge into a single distribution for coding, a "sign flipping" technique is used to sharpen the resulting distribution, thus, reducing the corresponding conditional entropy. The idea is that, before merging, the sign of errors whose distribution has negative estimated conditional mean are flipped, which can be mimicked by the decoder. Thus, twin distributions that are symmetric but of opposite sign, are merged at the "right" phase.

At the moment of writing, CALIC's executables can be downloaded from ftp://ftp.csd.uwo.ca/pub/from_wu.

V. JPEG-LS: High-Compression Performance at Low Complexity

While UCM and CALIC pushed the frontiers of lossless compressibility of continuous-tone images, the LOCO-I algorithm [55]–[57], developed in parallel to CALIC, showed

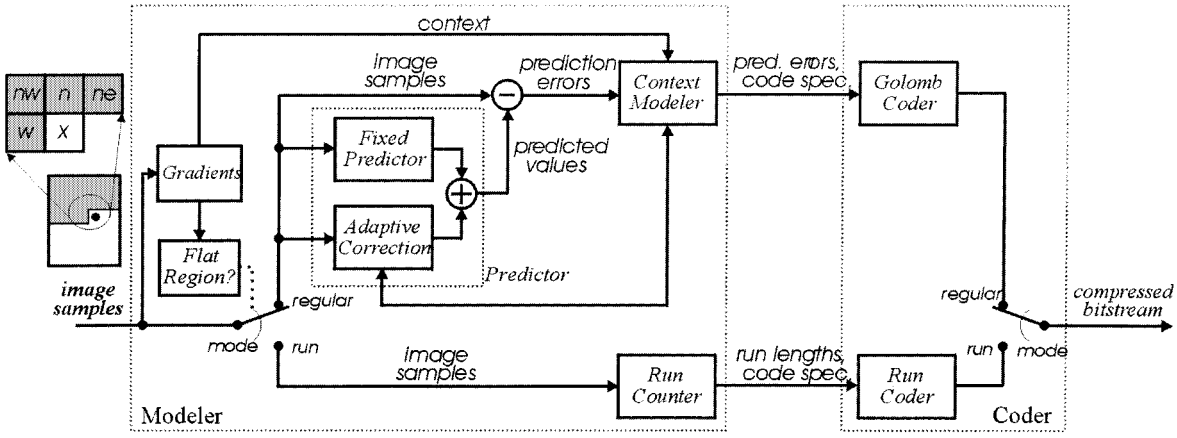

Fig. 4. JPEG-LS's encoder.

that low complexity and state-of-the-art compression are not incompatible, as had been suggested by FELICS. In many applications, a drastic complexity reduction can have more practical impact than a modest increase in compression. Since further attempts to improve on CALIC's compression ratios (see, e.g., [61]) confirmed that a point of diminishing returns was being reached, the alternative of applying judicious modeling to obtain competitive compression at significantly lower complexity levels seems appealing. Rather than pursuing the optimization of the image modeling principles of UCM, the main objective driving the design of LOCO-I (LOw COmplexity LOssless COmpression for Images) is to systematically "project" these principles into a low-complexity plane, both from a modeling and coding perspective. Thus, LOCO-I differs from FELICS in that it follows a more traditional predictor-modeler-coder structure along the paradigm of [50] and [53].

In 1994, the JPEG committee solicited proposals for a new international standard for continuous-tone lossless image compression [18]. The Call for Contributions was answered by eight industrial and academic organizations that submitted a total of nine proposals, including CALIC and LOCO-I. Two proposals were based on reversible transform coding, while the others built on context or block-based, adaptive, predictive coding. As a result of the standardization process, LOCO-I is the algorithm at the core of the new standard, termed JPEG-LS [16]. It was selected due to its optimal placement in a conceptual compression/complexity curve, within a few percentage points of the best available compression ratios (given by CALIC), but at a much lower complexity level.[2] The standard evolved after refinements of the algorithm introduced in [55]; here, we discuss the final scheme [56]. In Section VI we also comment on two other proposals, CREW [67], [13] and ALCM [45], [46], as these algorithms ended up impacting other standardization efforts.

Our description of JPEG-LS refers to the block diagram in Fig. 4 (from [56]), which includes the causal template used for prediction and modeling. The fixed component of the predictor switches among three simple predictors (I_w, I_n, and $I_w + I_n - I_{nw}$), resulting in a nonlinear function of the samples in the causal template, given by

$$\hat{x}_{\mathrm{MED}} \stackrel{\Delta}{=} \min(I_w, I_n, I_{nw}) + \max(I_w, I_n, I_{nw}) - I_{nw}.$$

This function (first suggested in [34]) is termed *median edge detector* (MED), as it incorporates prior knowledge through a rudimentary edge detection capability. The adaptive component of the predictor is limited to an *integer* additive term, analogous to an affine term in the adaptive predictor of [53]. It effects a context-dependent translation ("bias cancellation"), and can also be interpreted as part of the estimation procedure for the probabilistic model of the prediction residuals. The context dependence includes the sample ne, which is not used in the fixed part of the predictor.

The number of parameters per context is reduced to two by assuming a TSGD model for the prediction residuals. The integer part of the TSGD offset μ is canceled by the adaptive component of the predictor, which, through a simple addition-subtraction mechanism, is tuned to produce average residuals between -1 and 0, leaving a negative fractional shift s. Thus, the assumed distribution on the prediction error ϵ after bias cancellation decays as $\theta^{|\epsilon+s|}$, where $\theta \in (0, 1)$ and $s \in [0, 1)$. The choice of this interval for s is matched to the prefix codes used in the adaptive coding unit described below.

The context model is determined by quantized gradients as in [53]. Each of the differences $g_1 = I_{ne} - I_n$, $g_2 = I_n - I_{nw}$, and $g_3 = I_{nw} - I_w$, is quantized into up to nine connected regions by a quantizer $\kappa(\cdot)$. To preserve symmetry, the regions are indexed $-4, \cdots, -1, 0, 1, \cdots, 4$, with $\kappa(g) = -\kappa(-g)$, for a total of 729 different quantized context triplets. For a prediction residual ϵ, if the first nonzero element of a triplet $C = [q_1, q_2, q_3]$, where $q_j = \kappa(g_j)$, $j = 1, 2, 3$, is negative, the encoded value is $-\epsilon$, using context $-C$. This is anticipated by the decoder, which flips the sign if necessary to obtain the original error value. Merging contexts of "opposite signs" results in a total of 365 contexts. With two TSGD parameters per context, the total number of free statistical parameters is very manageable, and has proved to be a very effective compromise for a wide

[2]For example, it is reported in [56] that timing experiments with publicly available implementations yield about an 8 : 1 speed ratio on natural images and significantly more on compound documents.

range of image types and sizes. Notice also that, by using the template of Fig. 4 for prediction and modeling, JPEG-LS limits its image buffering requirement to one scan line.

In a low-complexity framework, the choice of a TSGD model is of paramount importance, since it leads to a very simple yet efficient coding unit. This unit derives from the family of optimal prefix codes for TSGDs characterized in [30], which are, in turn, based on the Golomb codes. Specifically, prediction residuals are encoded with codes from the family

$$\mathcal{C} = \{G_{2^k}(M(\cdot))|k \geq 0\} \cup \{G_1(M'(\cdot))\}$$

where G_m denotes the Golomb code of order m, $M(x)$ denotes the mapping from an integer x to its index in the interleaved sequence $0, -1, +1, -2, +2, \cdots$ (starting from index 0), and $M'(x) = M(-x-1)$. The use of the map M' in $\mathcal{C}$ reflects dependence on the TSGD parameter s. Codes in $\mathcal{C}-\{G_1(M'(\cdot))\}$ were first used for image compression applications in [36]; thus, the map M is often called *Rice mapping*.[3] In JPEG-LS, codes from $\mathcal{C}$ are adaptively selected with an on-line strategy reflecting the estimation of the parameters of the TSGD. The strategy turns out to be surprisingly simple, and it is derived using techniques presented in [29] and [44]. Specifically, the code parameter k is computed by the C programming language "one-liner"

$$\texttt{for}\,(\texttt{k} = 0;\ (\texttt{N} \ll \texttt{k}) < \texttt{A};\ \texttt{k++});$$

where N counts the number of prediction residuals that have been coded at that context, and A accumulates the magnitudes of the prediction residuals for that context. As a result, adaptive symbol-by-symbol coding is possible at very low complexity, thus, avoiding the use of the more complex arithmetic coders. The use of Golomb codes in conjunction with context modeling was pioneered in FELICS (see Section III). However, JPEG-LS uses a TSGD model, as opposed to the geometric distributions assumed in FELICS. Also, the above simple explicit formula for Golomb parameter estimation differs from the search procedure described in [15].

In order to address the redundancy of symbol-by-symbol coding in the low entropy range ("flat" regions), a major problem in FELICS, an alphabet extension is embedded in the JPEG-LS model ("run" mode). In Fig. 4, the switches labeled *mode* select operation in "regular" or "run" mode, as determined from the context by the simple "flatness" condition $g_1 = g_2 = g_3 = 0$. In run mode, the length of the run of the sample value I_w is adaptively coded using *block-MELCODE*, an adaptation technique for Golomb-type codes [51]. This specific adaptation technique is the most significant departure of JPEG-LS from the original LOCO-I.

In summary, the overall simplicity of JPEG-LS can be mainly attributed to its success in matching the complexity of the modeling and coding units, combining simplicity with the compression potential of context models, thus, "enjoying the best of both worlds." JPEG-LS implementations can be downloaded from http://www.hpl.hp.com/loco. Another JPEG-LS codec is available from ftp://dspftp.ece.ubc.ca/pub/jpeg-ls.

[3]Even though [12] precedes [36] by more than a decade, the prefix codes G_{2^k} are sometimes referred to as *Rice codes*.

VI. Other Approaches

LOCO-A and ALCM: An arithmetic coding version of LOCO-I, termed LOCO-A [56], is being specified as an extension of the baseline JPEG-LS standard [17]. This extension addresses the basic limitations that the standard presents when dealing with very compressible images, or images that are far from satisfying the assumptions underlying the model in JPEG-LS. LOCO-A is a natural extension of the JPEG-LS baseline, requiring the same buffering capability. It closes, in general, most of the (small) compression gap between JPEG-LS and CALIC, at the price of the additional computational complexity introduced by the arithmetic coder (but with no significant additional complexity for modeling). The basic idea behind LOCO-A follows from clustering contexts with similar conditional distributions into *conditioning states*, based on the value of the ratio A/N (used to determine the Golomb parameter k in JPEG-LS). The resulting state-conditioned distributions are arithmetic encoded, thus, relaxing the TSGD assumption, which is used only as a means to form the states. Here, A/N acts as a measure of activity level, discriminating between active areas (such as edges) and smooth regions, and can be seen as a refinement of the "error energy" used in CALIC. Clearly, LOCO-A applies the paradigm of [70], [54], using many contexts for prediction but only a few for coding.

Activity levels are also used in the activity level classification model (ALCM) algorithm [45], another contributor to the design of LOCO-A. Moreover, LOCO-A borrows from ALCM its binarization strategy, which differs from the one described for the Sunset family. The idea in ALCM is to apply the Rice mapping M, defined in Section V, to the prediction errors, and binarize the decisions using the code tree associated with a context-dependent Golomb code, with the context given by the activity level. A *binary* adaptive arithmetic code is used to encode the sequence of decisions. A key feature of ALCM is its adaptive predictor, based on six neighboring sample values, which is discussed in [46]. By emphasizing prediction accuracy, ALCM reported the best results on "near-lossless" compression among proposals submitted in response to [18]. In this *lossy* mode of operation, also standardized by JPEG-LS, every sample value in a reconstructed image component is guaranteed to differ from the corresponding value in the original image by up to a preset (small) amount, δ. Near-lossless performance is closely related to prediction accuracy, since the loss is introduced in a DPCM loop by quantizing the prediction residual into quantization bins of size $2\delta + 1$, with reproduction at the center of the interval.

CREW and Other Transform-Based Algorithms: CREW (compression with reversible embedded wavelets) uses an integer-to-integer wavelet transform, thus, allowing for embedded coding. Ideally, an embedded coder generates a bit stream that can be truncated at any point, and the resulting prefix can be decoded to reconstruct the original image with a fidelity approaching that of an "optimal" coder, tailored to produce the same bit rate as the prefix. With an

integer-to-integer transform, the end step of progressive decoding is a lossless representation. The wavelet model works reasonably well on natural images (although it falls short of the compression efficiency achieved by predictive schemes of similar complexity), but is not suited for images with sharp edges or text portions (compound documents, computer graphics). However, the novelty of the rich set of features provided by CREW triggered a new ISO standardization effort, JPEG 2000 [19]. This emerging standard, which draws heavily from [49], includes provisions for integer-to-integer wavelet transforms for lossless image compression, an approach pioneered by CREW and SPIHT [43]. Other schemes for progressive image transmission are described in [5] and [62].[4]

The transform step in transform-based schemes can be viewed as the computation of "prediction residuals" from a large, *noncausal* neighborhood. Unlike sophisticated predictive schemes, this "predictor" is nonadaptive and linear, reflecting a "prior belief" on a more restricted model.[5] The context model for encoding the transform coefficients is based on neighboring coefficients, which is analogous to the use of neighboring prediction errors in, e.g., Sunset, but differs from UCM, CALIC, and JPEG-LS. The progression by quality is usually given by the encoding of the coefficients by bit planes. This aspect is analogous to the binarization process in, e.g., Sunset and ALCM. However, in predictive schemes, all the bits from one sample are encoded before proceeding with the next sample. In contrast, the context model in transform-based schemes can include bits already encoded from noncausal neighbors. The interaction between context modeling and transform coding warrants further investigation.

TMW—Two-Pass Mixing Modeling: The TMW algorithm [31] uses the two-pass modeling paradigm, in which a set of model parameters is estimated from the whole image in a first pass (*image analysis stage*), and is then used by an arithmetic coder, which codes the samples in a second pass (*coding stage*). The model parameters must be described to the decoder as header information. TMW achieves compression results marginally better than CALIC (the improvements reported in [31] rarely exceed 5%), at the price of being quite impractical due to its computational complexity.

The probability assignment in TMW can be seen as a mixture of parametric models. Mixtures are a popular tool for creating universal models, in the sense that the asymptotic performance of the mixture approaches that of each particular model in the mixture, even if the data agrees with that specific model (see, e.g., [42]). However, this theoretically appealing approach was not pursued in other state-of-the-art image compression algorithms. In TMW, the mixture is performed on variations of the t-distribution, which are coupled with linear *sample predictors* that combine the value of causal neighbors to determine the center of the distribution. The parameter of each t-distribution depends on past prediction errors. An alternative interpretation, along the lines of the modeling principles discussed in this paper, is that the distribution parameters are context-dependent, with the contexts given by past prediction errors (as in Sunset), and the dependency on the context being, in turn, parameterized (*sigma predictors*). This (second-level) parameterization allows for the use of large causal templates without significant penalty in model cost. In a sense, sigma predictors predict, from past errors, the expected error of sample predictors. As in UCM, the probability assigned to a sample value x results from integrating the continuous distribution in the interval $[x - 0.5, x + 0.5)$. The mixture coefficients are based on the (weighted) past performance of each model in a causal neighborhood (*blending predictors*), with weights that can be interpreted as "forgetting factors." The blending predictors yield an implicit segmentation of the image into segments, each "dominated" by a given sample predictor. The determination of mixture coefficients is in the spirit of *prediction with expert advice* [25], with each sample predictor and t-distribution acting as an "expert." The three-level hierarchy of linear predictors (sample, sigma, and blending predictors) is optimized iteratively by gradient descent in the image analysis stage.

It is worth noting that there is no inherent advantage to two-pass modeling, in the sense that the model cost associated with describing the parameters is known to be asymptotically equivalent to that resulting from the learning process in one-pass schemes [39] (see Section I). In fact, one-pass schemes adapt better to nonstationary distributions (TMW overcomes this problem by the locality of blending predictors and through the forgetting factor). Thus, it is the mixing approach that appears to explain the ability of TMW to adapt to a wide range of image types. While none of the modeling tools employed in TMW is essentially new, the scope of their optimization definitely is. Therefore, the limited impact that this optimization has on compression ratios as compared to, e.g., CALIC, seems to confirm that a point of diminishing returns is indeed being reached.

Heavily Adaptive Prediction Schemes: In addition to ALCM, other schemes build mainly on adaptive prediction. The ALPC algorithm, presented in [32] and [33], makes explicit use of local information to classify the context of the current sample, and to select a linear predictor that exploits the local statistics. The predictor is refined by gradient descent optimization on the samples collected in the selected cluster. ALPC collects statistics inside a window of previously encoded samples, centered at the sample to be encoded, and classifies all the samples in the window into clusters by applying the generalized Lloyd algorithm (LBG) on the samples' contexts. Then, the context is classified in one of the clusters and the most efficient predictor for that cluster is selected and refined via gradient descent. The prediction error is finally arithmetic coded. The compression results show that the algorithm performs well on textured and structured images, but has limitations on high-contrast zones. The computational complexity of this approach is very high.

[4]Notice that even on natural images, CALIC still outperforms the relatively complex ECECOW algorithm [62] or its highly complex variant [63], in lossless compression.

[5]Here, we are not concerned with the multiresolution aspect of scalability, which is given by the organization of coefficients into decomposition levels.

In [23], an edge adaptive prediction scheme that adaptively weights four directional predictors (the previous pixels in four directions), and an adaptive linear predictor, updated by gradient descent (Widrow–Hoff algorithm) on the basis of local statistics, are used. The weighting scheme assumes a Laplacian distribution for the prediction errors, and uses a Bayesian weighting scheme based on the prediction errors in a small window of neighboring samples.

A covariance-based predictor that adapts its behavior based on local covariance estimated from a causal neighborhood is presented in [24]. This predictor tracks spatially varying statistics around edges and can be the basis of a lossless image coder that achieves compression results comparable to CALIC's. The high computational complexity of this approach can be reduced by using appropriate heuristics, and the price paid in terms of compression is reported to be negligible.

LZ-Based Schemes: As discussed in Section I, *universal* compression tools based on the LZ algorithm do not perform well when directly applied to natural images. However, the LZ algorithm has been adapted for image compression in various forms. The popular file format PNG achieves lossless compression through prediction (in two passes) and a variant of LZ1 [68]. A generalization of LZ2 [69] to image compression, with applications in lossy and lossless image coding, is presented in [7] and [8], and refined in [41]. At each step, the algorithm selects a point of the input image (also called growing point). The encoder uses a match heuristic to decide which block of a local dictionary (that stores a constantly changing set of vectors) is the best match for the subblock anchored at the growing point. The index of the block is transmitted to the decoder, a dictionary update heuristic adds new blocks to the dictionary, and the new growing point is selected. Decompression is fast and simple; once an index is received, the corresponding block is copied at the growing point. A similar approach was presented in [11]. Two-dimensional generalizations of LZ1 are presented in [47] and [48].

Block Coding: Block coding is sometimes used to exploit the correlation between symbols within a block (see, e.g., [71]), as the per-symbol entropy of the blocks is a decreasing function of their length. Notice that, if we ignore the complexity axis, the main theorem in [40] shows that this strategy for achieving compression ratios corresponding to higher order entropies is inferior to one based on context conditioning. However, block coding can be convenient in cases where fast decoding is of paramount importance.

VII. Color and Palletized Images

Many modern applications of lossless image compression deal mainly with color images, with a degree of correlation between components depending on the color space. Other images, e.g., satellite, may have hundreds of bands. In this section we discuss how the tools presented for single-component images are integrated for the compression of multiple-component images.

A simple approach consists of compressing each component independently. This does not take into account the correlation that often links the multiple bands, which could be exploited for better compression. For example, the JPEG-LS syntax supports both interleaved and noninterleaved (i.e., component by component) modes, but even in the interleaved modes, possible correlation between color planes is limited to sharing statistics, collected from all planes. In particular, prediction and context determination are performed separately for each component.

For some color spaces (e.g., RGB), good decorrelation can be obtained through simple lossless color transforms as a preprocessing step. For example, using JPEG-LS to compress the (R-G,G,B-G) representation of a set of photographic images, with suitable modular reduction applied to the differences [56], yields savings between 25% and 30% over compressing the respective (R,G,B) representation. Given the variety of color spaces, the standardization of specific filters was considered beyond the scope of JPEG-LS, and color transforms are expected to be handled at the application level.

A multiple-component version of CALIC is presented in [64]. The decision of switching to binary mode takes into account also the behavior of the corresponding samples in a reference component. In continuous-tone mode, the predictor switches between interband and intraband predictors, depending on the value of an interband correlation function in a causal template. Interband prediction extends GAP to take into account the values of the corresponding samples in the reference component. Context modeling is done independently for the inter- and intraband predictors, as they rely on different contexts. Interband context modeling depends on a measure of activity level that is computed from the reference component, and on previous errors for the left and upper neighbors in the current component.

Similarly, a multiple-component scheme akin to LOCO-I/JPEG-LS, termed SICLIC, is presented in [4]. A reference component is coded as in JPEG-LS. In regular mode, two coders work in parallel, one as in JPEG-LS, and the other using information from the reference component. For the latter, prediction and context determination are based on the differences between sample values and their counterpart in the reference component. After the adaptive prediction correction step, SICLIC chooses between intra- and interband prediction based on the error prediction magnitude accumulated at the context for both methods. Coding is performed as in JPEG-LS. In run mode, further savings are obtained by checking whether the same run occurs also in the other components. It is reported in [56] that the results obtained by SICLIC are, in general, similar to those obtained when JPEG-LS is applied after the above color transforms. However, SICLIC *does not* assume prior knowledge of the color space.

Palletized images, popular on the Internet, have a single component, representing an array of indices to a palette table, rather than multiple components as in the original color space representation. Indices are just labels for the colors they represent, and their numeric value may bear no relation to the color components. Furthermore, palletized images often contain combinations of synthetic, graphic, and text bitmap data, and might contain a sparse color set. Therefore, many

of the assumptions for continuous-tone images do not hold for the resulting arrays of indices, and palletized images may be better compressed with algorithms specifically designed for them (see, e.g., [1]–[3]). Also, LZ-type algorithms, such as PNG (without prediction), do reasonably well, especially on synthetic/graphic images. However, algorithms for continuous-tone images can often be advantageously used after an appropriate reordering of the palette table, especially for palletized natural images. The problem of finding the optimal palette ordering for a given image and compression algorithm is known to be computationally hard (see, e.g., [66], [27]). Some heuristics are known that produce good results at low complexity, without using image statistics. For example, [66] proposes to arrange the palette colors in increasing order of luminance value, so that samples that are close in space in a smooth image, and tend to be close in color and luminance, will also be close in the index space. The JPEG-LS data format provides tools for encoding palletized images in an appropriate index space. To this end, each decoded sample value (e.g., and 8-bit index) can be mapped to a reconstructed sample value (e.g., an RGB triplet) by means of mapping tables. According to [56], with the simple reordering of [66], JPEG-LS outperforms PNG by about 6% on palletized versions of common natural images. On the other hand, PNG may be advantageous for dithered, half-toned, and some synthetic/graphic images for which LZ-type methods are better suited.

VIII. Experimental Results and Prospects

The JPEG-LS standard for lossless compression of continuous-tone images is the state-of-the-art in terms of compression and computational efficiency. The standard achieves compression ratios close to the best available on a broad class of images; it is very fast and computationally simple. CALIC remains the benchmark for compression efficiency. Although considerably more complex than JPEG-LS, it is still suitable for many applications.

Table 1, extracted from [56], shows lossless compression results of JPEG-LS and CALIC, averaged over independently compressed color planes, on the subset of 8-bit/sample images from the benchmark set provided in the Call for Contributions leading to the JPEG-LS standard [18]. This set includes a wide variety of images, such as compound documents, aerial photographs, scanned, and computer-generated images. The table also includes the compression results of FELICS and the arithmetic coding version of lossless JPEG, as representatives of the previous generation of coders.

Other algorithms (e.g., TMW, ALPC) have achieved compression performance comparable to CALIC (or marginally better on some data), but remain impractical due to their high computational complexity.

Extensive comparisons on medical images of various types are reported in [6], where it is recommended that the Digital Imaging and Communications in Medicine (DICOM) standard add transfer syntaxes for both JPEG-LS and JPEG 2000.

Table 1
Compression Results On the JPEG-LS Benchmark Set (In Bits/Sample)

	L. JPEG	FELICS	JPEG-LS	CALIC
bike	3.92	4.06	3.63	3.50
cafe	5.35	5.31	4.83	4.69
woman	4.47	4.58	4.20	4.05
tools	5.47	5.42	5.08	4.95
bike3	4.78	4.67	4.38	4.23
cats	2.74	3.32	2.61	2.51
water	1.87	2.36	1.81	1.74
finger	5.85	6.11	5.66	5.47
us	2.52	3.28	2.63	2.34
chart	1.45	2.14	1.32	1.28
chart_s	3.07	3.44	2.77	2.66
compound1	1.50	2.39	1.27	1.24
compound2	1.54	2.40	1.33	1.24
aerial2	4.14	4.49	4.11	3.83
faxballs	0.84	1.74	0.90	0.75
gold	4.13	4.10	3.91	3.83
hotel	4.15	4.06	3.80	3.71
Average	3.40	3.76	3.19	3.06

Table 2
Compression Results On Color Images (In Bits/Sample)

	IB-CALIC	SICLIC	Filter+JPEG-LS
cats	1.81	1.86	1.85
water	1.51	1.45	1.45
cmpnd1	1.02	1.12	1.09
cmpnd2	0.92	0.97	0.96

Both schemes, in lossless mode, achieved an average compression performance within 2.5% of CALIC.

As for color images, Table 2 presents compression results, extracted from [4], for the multicomponent version of CALIC presented in [64] (denoted IB-CALIC) and for SICLIC, on a subset of 8-bit/sample RGB images from [18]. For comparison, JPEG-LS has been applied to the (R-G,G,B-G) representation of the images, with differences taken modulo 256 in the interval $[-128, 127]$ and shifted.

While the results of Table 2 suggest that, when the color space is known, interband correlation is adequately treated through simple filters, the advantage of interband schemes seems to reside in their robustness. For example, JPEG-LS compresses the (R-G,G,B-G) representation of the image *bike3* to 4.81 bits/sample, namely 10% *worse* than the nonfiltered representation. On the other hand, SICLIC achieves 4.41 bits/sample. Thus, SICLIC appears to exploit interband correlation whenever this correlation exists, but does not deteriorate significantly in case of uncorrelated color planes.

Table 3 presents compression results obtained with JPEG-LS[6] and CALIC[7] (with default parameters) on five other, 8-bit/sample, images.[8] These algorithms do not appear to perform as desired when compared to a popular universal compression tool, *gzip*, which is based on the LZ1 algorithm [68].

[6]from http://www.hpl.hp.com/loco

[7]from ftp://ftp.csd.uwo.ca/pub/from_wu

[8]from http://links.uwaterloo.ca/BragZone

Table 3
Compression Results On Five Other Images (In Bits/Sample)

	gzip -9	JPEG-LS	CALIC
france	0.34	1.41	0.82
frog	3.83	6.04	5.85
library	4.76	5.10	5.01
mountain	5.27	6.42	6.26
washsat	2.68	4.13	3.67

The images are as follows.

- france: $[672 \times 496]$—A grayscale version of an overhead slide as might be found in a business presentation. It contains text and graphics superimposed on a gradient background.
- frog: $[621 \times 498]$—Odd-shaped, dithered image picturing a frog.
- library: $[464 \times 352]$—Image scanned from a student handbook showing pictures of on-campus libraries. The pictures have a dithered quality, typical of many inexpensive publications.
- mountain: $[640 \times 480]$—A high-contrast landscape photograph.
- washsat: $[512 \times 512]$—A satellite photograph of the Washington, DC area.

A more appropriate usage of the compression tool can often improve the compression results. This is indeed the case with *mountain* and *washsat* in the above image set, as these pictures use few gray levels. A simple histogram compaction (in the form of a mapping table) results in a dramatic improvement: the compression ratios for JPEG-LS become, respectively, 5.22 and 2.00 bits/sample (5.10 and 2.03 bits/sample using CALIC). This histogram compaction is done off-line, and is supported by the JPEG-LS syntax through mapping tables. The standard extension (JPEG-LS Part 2 [17]) includes provisions for on-line compaction.

Nevertheless, *gzip* also does better than JPEG-LS and CALIC on the other three images. The poor performance of JPEG-LS and CALIC on *frog* and *library* seems to be caused by the dithered quality of the pictures and the failure of the prediction step.[9] In the case of the synthetic image *france*, its gradual shading prevents efficient use of the run mode in JPEG-LS, and may not be suited to CALIC's binary mode.

As JPEG-LS, CALIC, and all the previously cited lossless image compression algorithms build on certain assumptions on the image data, they may perform poorly in case these assumptions do not hold. While it may be argued that the current image models are unlikely to be improved significantly, it cannot be expected from a single algorithm to optimally handle all types of images. In particular, there is evidence that on synthetic or highly compressible images there is still room for improvement. The field of lossless image compression is therefore open to fruitful research.

[9]Compression results on *frog* improve by 15% after histogram compaction, but still fall significantly short of *gzip*.

ACKNOWLEDGMENT

B. Carpentieri would like to thank J.-C. Grossetie for his precious collaboration during the European Commission contracts devoted to the assessment of the state of the art of lossless image coding, and also the other participants to the same contracts: G. Ciscar, D. Marini, D. Saupe, J.-P. D'Ales, M. Hohenadel and P. Pagny. Special thanks also to J. A. Storer, G. Motta, and F. Rizzo for useful discussions.

REFERENCES

[1] P. J. Ausbeck Jr., "Context models for palette images," in *Proc. 1998 Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 309–318.

[2] ——, "A streaming piecewise-constant model," in *Proc. 1999 Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 208–217.

[3] ——, "The skip-innovation model for sparse images," in *Proc. 2000 Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 43–52.

[4] R. Barequet and M. Feder, "SICLIC: A simple inter-color lossless image coder," in *Proc. 1999 Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 501–510.

[5] A. Bilgin, P. Sementilli, F. Sheng, and M. W. Marcellin, "Scalable image coding with reversible integer wavelet transforms," *IEEE Trans. Image Process.*, vol. 9, pp. 1972–1977, Nov. 2000.

[6] D. Clunie, "Lossless compression of grayscale medical images—Effectiveness of traditional and state of the art approaches," *Proc. SPIE (Medical Imaging)*, vol. 3980, Feb. 2000.

[7] C. Constantinescu and J. A. Storer, "On-line adaptive vector quantization with variable size codebook entries," in *Proc. 1993 Data Compression Conf.*, Snowbird, UT, Mar. 1993, pp. 32–41.

[8] ——, "Improved techniques for single-pass adaptive vector quantization," *Proc. IEEE*, vol. 82, pp. 933–939, June 1994.

[9] M. Feder and N. Merhav, "Relations between entropy and error probability," *IEEE Trans. Inform. Theory*, vol. 40, pp. 259–266, Jan. 1994.

[10] R. Gallager and D. V. Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 228–230, Mar. 1975.

[11] J. M. Gilbert and R. W. Brodersen, "A lossless 2-D image compression technique for synthetic discrete-tone images," in *Proc. 1998 Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 359–368.

[12] S. W. Golomb, "Run-length encodings," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 399–401, July 1966.

[13] M. Gormish, E. Schwartz, A. Keith, M. Boliek, and A. Zandi, "Lossless and nearly lossless compression of high-quality images," *Proc. SPIE*, vol. 3025, pp. 62–70, Mar. 1997.

[14] P. G. Howard, "The design and analysis of efficient data compression systems," Ph.D. dissertation, Dept. Comput. Sci., Brown Univ., 1993.

[15] P. G. Howard and J. S. Vitter, "Fast and efficient lossless image compression," in *Proc. 1993 Data Compression Conf.*, Snowbird, UT, Mar. 1993, pp. 351–360.

[16] ISO/IEC, "Information technology—Lossless and near-lossless compression of continuous-tone still images,", 14 495-1, ITU Recommendation T.87, 1999.

[17] ISO/IEC, "JPEG-LS Part 2,", FCD 14 495-2, Apr. 2000.

[18] ISO/IEC, "Call for contributions—Lossless compression of continuous-tone still images,", JTC1/SC29/WG1, Mar. 1994.

[19] ISO/IEC, "Information technology—JPEG 2000 image coding system,", JTC1/SC29/WG1 FCD15444-1, Mar. 2000.

[20] G. G. Langdon Jr., "Sunset: A hardware oriented algorithm for lossless compression of gray scale images," *Proc. SPIE*, vol. 1444, pp. 272–282, Mar. 1991.

[21] G. G. Langdon Jr. and M. Manohar, "Centering of context-dependent components of prediction error distributions," *Proc. SPIE (Applications of Digital Image Processing XVI)*, vol. 2028, pp. 26–31, July 1993.

[22] G. G. Langdon Jr. and C. A. Haidinyak, "Experiments with lossless and virtually lossless image compression algorithms," *Proc. SPIE*, vol. 2418, pp. 21–27, Feb. 1995.

[23] W. S. Lee, "Edge adaptive prediction for lossless image coding," in *Proc. 1999 Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 483–490.

[24] X. Li and M. Orchard, "Edge-directed prediction for lossless compression of natural images," in *Proc. 1999 Int. Conf. Image Processing*, Oct. 1999.
[25] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Inform. Comput.*, vol. 108, no. 2, pp. 212–261, 1994.
[26] N. D. Memon and K. Sayood, "Lossless image compression: A comparative study," *Proc. SPIE (Still-Image Compression)*, vol. 2418, pp. 8–20, Feb. 1995.
[27] N. D. Memon and A. Venkateswaran, "On ordering color maps for lossless predictive coding," *IEEE Trans. Image Process.*, vol. 5, pp. 1522–1527, Nov. 1996.
[28] N. Merhav, M. Feder, and M. Gutman, "Some properties of sequential predictors for binary Markov sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 887–892, May 1993.
[29] N. Merhav, G. Seroussi, and M. J. Weinberger, "Coding of sources with two-sided geometric distributions and unknown parameters," *IEEE Trans. Inform. Theory*, vol. 46, pp. 229–236, Jan. 2000.
[30] ——, "Optimal prefix codes for sources with two-sided geometric distributions," *IEEE Trans. Inform. Theory*, vol. 46, pp. 121–135, Jan. 2000.
[31] B. Meyer and P. Tischer, "TMW—A new method for lossless image compression," in *Proc. 1997 Int. Picture Coding Symp. (PCS97)*, Berlin, Germany, Sept. 1997.
[32] G. Motta, J. A. Storer, and B. Carpentieri, "Adaptive linear prediction lossless image coding," in *Proc. 1999 Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 491–500.
[33] ——, "Lossless image coding via adaptive linear prediction and classification," *Proc. IEEE*, pp. 1790–1796, Nov. 2000.
[34] H. Murakami, S. Matsumoto, Y. Hatori, and H. Yamamoto, "15/30 Mbit/s universal digital TV codec using a median adaptive predictive coding method," *IEEE Trans. Commun.*, vol. 35, pp. 637–645, June 1987.
[35] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand, 1993.
[36] R. F. Rice, "Some practical universal noiseless coding techniques," Jet Propulsion Laboratory, Pasadena, CA, Tech. Rep. JPL-79-22, Mar. 1979.
[37] J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM J. Res. Develop.*, vol. 20, pp. 198–203, May 1976.
[38] ——, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 629–636, July 1984.
[39] ——, *Stochastic Complexity in Statistical Inquiry*. New Jersey/London: World Scientific, 1989.
[40] J. Rissanen and G. G. Langdon Jr., "Universal modeling and coding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 12–23, Jan. 1981.
[41] F. Rizzo, J. A. Storer, and B. Carpentieri, "Improving single-pass adaptive VQ," in *Proc. 1999 Int. Conf. Acoustic, Speech, and Signal Processing*, Mar. 1999.
[42] B. Ryabko, "Twice-universal coding," *Prob. Inform. Trans.*, vol. 20, pp. 173–177, July–Sept. 1984.
[43] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1303–1310, Sept. 1996.
[44] G. Seroussi and M. J. Weinberger, "On adaptive strategies for an extended family of Golomb-type codes," *Proc. 1997 Data Compression Conf.*, pp. 131–140, Mar. 1997.
[45] D. Speck, "Activity level classification model (ALCM),", A proposal submitted in response to the Call for Contributions for ISO/IEC JTC 1.29.12, 1995.
[46] ——, "Fast robust adaptation of predictor weights from min/max neighboring pixels for minimum conditional entropy," in *Proc. 29th Asilomar Conf. Signals, Systems, and Computers*, Asilomar, CA, Nov. 1995, pp. 234–238.
[47] J. A. Storer, "Lossless image compression using generalized LZ1-type methods," in *Proc. 1996 Data Compression Conf.*, Snowbird, UT, Mar. 1996, pp. 290–299.
[48] J. A. Storer and H. Helfgott, "Lossless image compression by block matching," *Comput. J.*, vol. 40, no. 2/3, pp. 137–145, 1997.
[49] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, pp. 1158–1170, July 2000.
[50] S. Todd, G. G. Langdon Jr., and J. Rissanen, "Parameter reduction and context selection for compression of the gray-scale images," *IBM J. Res. Develop.*, vol. 29, pp. 188–193, Mar. 1985.
[51] I. Ueno and F. Ono, "Proposed modification of LOCO-I for its improvement of the performance,", ISO/IEC JTC1/SC29/WG1 doc. N297, Feb. 1996.
[52] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34-4, pp. 30–44, Apr. 1991.
[53] M. J. Weinberger, J. Rissanen, and R. B. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. Image Process.*, vol. 5, pp. 575–586, Apr. 1996.
[54] M. J. Weinberger and G. Seroussi, "Sequential prediction and ranking in universal context modeling and data compression," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1697–1706, Sept. 1997.
[55] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity, context-based, lossless image compression algorithm," in *Proc. 1996 Data Compression Conf.*, Snowbird, UT, Mar. 1996, pp. 140–149.
[56] ——, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, pp. 1309–1324, Aug. 2000.
[57] ——, "From LOCO-I to the JPEG-LS standard," in *Proc. 1999 Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999.
[58] I. H. Witten, A. Moffat, and T. Bell, *Managing Gigabytes*. New York: Van Nostrand, 1994.
[59] I. H. Witten, R. Neal, and J. M. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.
[60] X. Wu, "An algorithmic study on lossless image compression," in *Proc. 1996 Data Compression Conf.*, Snowbird, UT, Mar. 1996, pp. 150–159.
[61] ——, "Efficient lossless compression of continuous-tone images via context selection and quantization," *IEEE Trans. Image Process.*, vol. 6, pp. 656–664, May 1997.
[62] ——, "High order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," in *Proc. 31st Asilomar Conf. Signals, Systems, and Computers*, Asilomar, CA, Nov. 1997, pp. 1378–1382.
[63] ——, "Context quantization with Fisher discriminant for adaptive embedded image coding," in *Proc. 1999 Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 102–111.
[64] X. Wu, W. Choi, and N. D. Memon, "Lossless interframe image compression via context modeling," in *Proc. 1998 Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 378–387.
[65] X. Wu and N. D. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, pp. 437–444, Apr. 1997.
[66] A. Zaccarin and B. Liu, "A novel approach for coding color quantized images," *IEEE Trans. Image Process.*, vol. 2, pp. 442–453, Oct. 1993.
[67] A. Zandi, J. Allen, E. Schwartz, and M. Boliek, "CREW: Compression with reversible embedded wavelets," in *Proc. 1995 Data Compression Conf.*, Snowbird, UT, Mar. 1995, pp. 351–360.
[68] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 337–343, May 1977.
[69] ——, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 530–536, Sept. 1978.
[70] M. J. Weinberger and G. Seroussi, "On the interaction between universal context modeling and prediction," in *1994 IEEE Int. Symp. Inform. Theory*, Trondheim, Norway, July 1994, p. 385.
[71] B. Gandhi, C. Honsinger, M. Rabbani, and C. Smith, "Differential Adaptive Run Coding (DARC),", A proposal submitted in response to the Call for Contributions 1396 for ISO/IEC.

Bruno Carpentieri (Member, IEEE) received the "Laurea" degree in computer science from the University of Salerno, Salerno, Italy, and the M.A. and Ph.D. degrees in computer science from Brandeis University, Waltham, MA.

Since 1991, he has been Assistant Professor of Computer Science (Ricercatore) at the University of Salerno. His research interests include lossless image compression, video compression and motion estimation, information hiding, parallel computing, and theory of computation. He was co-chair, in 1997, of the International Conference on Compression and Complexity of Sequences, and, since then, he has been a program committee member of the IEEE Data Compression Conference. He has been responsible for two European Commission contracts regarding the report on the state-of-the-art of lossless image compression.

Marcelo J. Weinberger (Senior Member, IEEE) was born in Buenos Aires, Argentina, in 1959. He received the B.Sc. degree in electrical engineering from the Universidad de la República, Montevideo, Uruguay, in 1983, and the M.Sc. and D.Sc. degrees in electrical engineering from Technion - Israel Institute of Technology, Haifa, Israel, in 1987 and 1991, respectively.

From 1985 to 1992, he was with the Department of Electrical Engineering, Technion, joining the faculty for the 1991–1992 academic year. During 1992–1993, he was a Visiting Scientist at IBM Almaden Research Center, San Jose, CA. Since 1993, he has been with Hewlett-Packard Laboratories, Palo Alto, CA. His research interests are in information theory and statistical modeling, including source coding, image compression, and sequential decision problems. He is a co-inventor of the image compression algorithm at the core of the JPEG-LS standard, and was an Editor of the standard specification. He also contributed to the coding algorithm of the emerging JPEG2000 standard.

Dr. Weinberger is an Associate Editor for Source Coding for the IEEE TRANSACTIONS ON INFORMATION THEORY.

Gadiel Seroussi (Fellow, IEEE) was born in Montevideo, Uruguay, in 1955. He received the B.Sc. degree in electrical engineering and the M.Sc. and D.Sc. degrees in computer science from Technion - Israel Institute of Technology, Haifa, Israel, in 1977, 1979, and 1981, respectively.

From 1981 to 1987, he was with the Faculty of the Computer Science Department, Technion. During the 1982–1983 academic year, he was a Postdoctoral Fellow at the Mathematical Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, NY. From 1986 to 1988, he was a Senior Research Scientist with Cyclotomics, Inc., Berkeley, CA. Since 1988, he has been with Hewlett-Packard Laboratories, Palo Alto, CA, where he leads the Information Theory Research group. His research interests include the mathematical foundations and practical applications of information theory, error correcting codes, data and image compression, and cryptography. He has published numerous journal and conference articles in these areas, and is a coauthor of the recent book *Elliptic Curves in Cryptography*. He is a co-inventor of the image compression algorithm at the core of the JPEG-LS standard, and was on the editorial committee for the standard specification. He also contributed to the coding algorithm of the emerging JPEG2000 standard.