Lossless Generalized-LSB Data Embedding

Mehmet Utku Celik, *Student Member, IEEE*, Gaurav Sharma, *Senior Member, IEEE*, Ahmet Murat Tekalp, *Fellow, IEEE*, and Eli Saber, *Senior Member, IEEE*

***Abstract*—We present a novel lossless (reversible) data-embedding technique, which enables the exact recovery of the original host signal upon extraction of the embedded information. A generalization of the well-known least significant bit (LSB) modification is proposed as the data-embedding method, which introduces additional operating points on the capacity-distortion curve. Lossless recovery of the original is achieved by compressing portions of the signal that are susceptible to embedding distortion and transmitting these compressed descriptions as a part of the embedded payload. A prediction-based conditional entropy coder which utilizes unaltered portions of the host signal as side-information improves the compression efficiency and, thus, the lossless data-embedding capacity.**

***Index Terms*—Arithmetic coding, conditional entropy coding, context modeling, data embedding, data hiding, least significant bit (LSB) modification, watermark.**

I. Introduction

MULTIMEDIA data embedding, or digital watermarking,[1] refers to the process of inserting information bits into a host multimedia signal without introducing perceptible artifacts [1]–[3]. A variety of embedding techniques, ranging from high-capacity bit modification to transform-domain spread spectrum methods, are used in various applications such as authentication [4], [5], meta-data tagging, content-protection, and secret communications.

Most multimedia data-embedding techniques modify and, hence, distort the host signal in order to insert the additional information (other techniques may encode this information into the "representation" of the signal, e.g., color space). The distortion induced on the host signal by the data-embedding technique is called the *embedding distortion*. Often, the embedding distortion is small, yet irreversible, i.e., it cannot be removed to recover the original host signal. In many applications, the loss of host signal fidelity is not prohibitive as long as original and modified signals are perceptually equivalent. However, in a number of domains—such as military, legal and medical imaging—although some embedding distortion is admissible, permanent loss of signal fidelity is undesirable. This highlights the need for *lossless data embedding*[2] techniques. These techniques, like their lossy counterparts, insert information bits by modifying the host signal, thus inducing an embedding distortion. Nevertheless, they also enable the removal of such distortions and the exact—lossless—restoration of the original host signal after extraction of embedded information. Particular applications include embedding of DICOM header [6] information into medical images and providing fragile authentication watermarking for aerial/surveillance images [7].

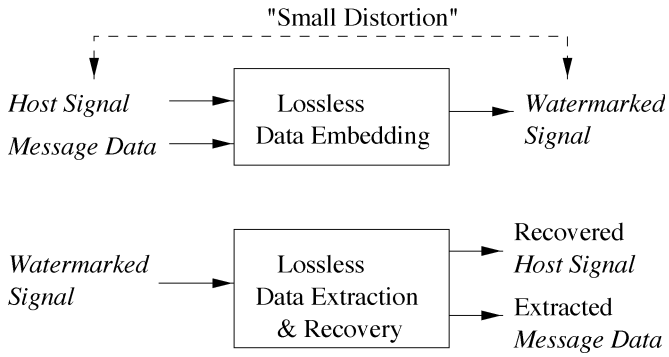

Fig. 1. Lossless data embedding, extraction, and recovery.

A general block diagram representing lossless data-embedding schemes is seen in Fig. 1. The lossless embedding step takes the host signal and the message data and produces a watermarked signal in which the message data is embedded. The data extraction and recovery process uses the watermarked signal to extract the embedded data and to recover the original host signal exactly. Note that though the recovery process allows reconstruction of the original host signal with no distortion, it is still desirable to keep the embedding distortion, i.e., the difference between the host and watermarked signal, to a minimum so that applications that do not have access to the extraction and recovery process do not incur a heavy penalty in image quality.

Lossless data-embedding techniques may be classified into one of the following two categories: Type-I algorithms [8], [9] employ additive spread spectrum techniques, where a spread spectrum signal corresponding to the information payload is superimposed on the host in the embedding phase. At the decoder, detection of the embedded information is followed by a restoration step where the watermark signal is removed,

Manuscript received July 17, 2002; revised March 17, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gopal Pingali.

M. U. Celik is with the Electrical and Computer Engineering Department, University of Rochester, Rochester, NY 14627-0126 USA (e-mail: celik@ece.rochester.edu).

G. Sharma is with the Electrical and Computer Engineering Department and the Department of Biostatistics and Computational Biology, University of Rochester, Rochester, NY 14627-0126 USA (e-mail: gaurav.sharma@rochester.edu).

A. M. Tekalp is with College of Engineering, Koc University, Istanbul, Turkey, and also with the Department of Electrical and Computer Engineering, University of Rochester, NY 14627-0126 USA (e-mail: tekalp@ece.rochester.edu).

E. Saber is with Xerox Corporation, Webster, NY 14580 USA, and also with the Department of Electrical and Computer Engineering, University of Rochester, NY 14627-0126 USA (e-mail: eli.saber@usa.xerox.com).

Digital Object Identifier 10.1109/TIP.2004.840686

[1]Throughout this paper, the terms *data embedding*, *digital watermarking*, and *watermarking* are used interchangeably and subtle nuances among the terms are ignored.

[2]In the literature, lossless data embedding is also referred as *reversible*, *invertible*, or *distortion-free* data embedding.

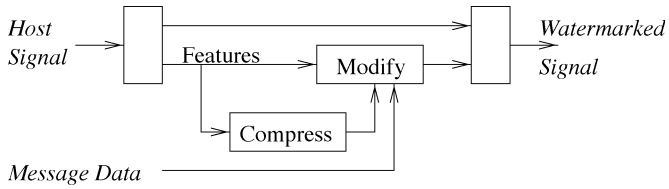

Fig. 2. Type-II lossless data embedding.

i.e., subtracted, to restore the original host signal. Potential reversibility problems associated with the limited range of values in the digital representation of the host signal, e.g., overflows and underflows during addition and subtraction, are prevented by adopting modulo arithmetic [8] or a circular interpretation of the bijective transform [9]. Due to their spread-spectrum nature, Type-I algorithms are robust with regard to the data embedding and allow for extraction of message data even if the host signal is perturbed (within reasonable limits) prior to detection. The original host signal, however, cannot be accurately recovered once the watermarked signal is perturbed. While the robustness is often desirable, modulo arithmetic typically produces disturbing salt-and-pepper artifacts in the watermarked signal compromising the desired goal of minimizing embedding distortion.

In Type-II algorithms [10]–[13], information bits are embedded by modifying, e.g., overwriting, selected features (portions) of the host signal—for instance least significant bits, high-frequency wavelet coefficients. In this class of algorithms, the embedding function is irreversible. Recovery of the original host is achieved by compressing the original features and transmitting the compressed bit stream as a part of the embedded payload. At the decoder, the embedded payload—including the compressed bit stream—is extracted, and the original host signal is restored by replacing the modified features with the decompressed original features. A general block diagram representing Type-II algorithms is seen in Fig. 2. In general, Type-II algorithms do not cause salt-and-pepper artifacts in the watermarked signal and can facilitate higher embedding capacities, albeit at the loss of the robustness of the first group.

This paper presents a high-capacity, low-distortion, Type-II lossless data-embedding algorithm. First, in Section II, we introduce a generalization of the well-known least significant bit (LSB) modification method as the underlying data-embedding technique. This technique modifies the lowest levels—instead of bit planes—of the host signal to accommodate the payload information. This generalization has a finer capacity-distortion granularity. In the second part, Section III, a lossless data-embedding algorithm for continuous-tone images is built based on the generalized LSB modification method. This spatial domain algorithm modifies the lowest levels of the raw pixel values as signal features. As in all Type-II algorithms, recovery of the original image is enabled by compressing, transmitting, and recovering these features. However, unlike in existing Type-II algorithms, the novel feature compression step utilizes the rest of the host signal as side-information. This property of the proposed method provides excellent compression of the image features. Earlier algorithms in the literature [10], [11] tend to select more complex features to improve the compression performance—thus, the lossless-embedding capacity. In Section III-B, the embedding capacity-distortion performance of the algorithm is further improved by modifying only a selected subset of signal samples. Finally, a simple capacity-distortion control mechanism, which minimizes the embedding distortion for a given target capacity, is developed.

II. Generalized-LSB (G-LSB) Embedding

One of the earliest data-embedding methods is the LSB modification. In this well-known method, the LSB of each signal sample is replaced (over written) by a payload data bit embedding one bit of data per input sample. If additional capacity is required, two or more LSBs may be over written allowing for a corresponding bits per sample. During extraction, these bits are read in the same scanning order, and payload data is reconstructed. LSB modification is a simple, nonrobust embedding technique with a high-embedding capacity and small bounded-embedding distortion (± 1). A generalization of the LSB-embedding method, namely G-LSB, is employed here. If the host signal is represented by a vector $\mathbf{s}$, the G-LSB embedding and extraction processes can be represented as

$$\mathbf{s}_w = Q_L(\mathbf{s}) + \mathbf{w} \tag{1}$$

$$\mathbf{w} = \mathbf{s}_w - Q_L(\mathbf{s}_w) = \mathbf{s}_w - Q_L(\mathbf{s}) \tag{2}$$

where $\mathbf{s}_w$ represents the signal containing the embedded information, $\mathbf{w}$ represents the embedded payload vector of L-ary symbols, i.e., $w_i \in \{0, 1, \ldots, L-1\}$, and

$$Q_L(x) = L \left\lfloor \frac{x}{L} \right\rfloor \tag{3}$$

is an L-level scalar quantization function, and $\lfloor\ \rfloor$ represents the operation of truncation to the integer part.

In the embedding phase, the lowest L levels of the signal samples are replaced (over-written) by the watermark payload using a quantization step followed by an addition. During extraction, the watermark payload is extracted by obtaining the quantization error—or simply reading lowest L levels—of the watermarked signal. The classical LSB modification, which embeds a binary symbol (bit) by overwriting the least significant bit of a signal sample, is a special case where $L = 2$. G-LSB embedding enables embedding of noninteger number of bits in each signal sample and, thus, introduces new operating points along the rate (capacity)-distortion curve.

A. Binary to L-ary (L-ary to Binary) Conversion

In the preceding section, we assumed that the watermark payload is presented as a string of L-ary symbols w_i. In typical practical applications payload data is input and output as binary strings. Therefore, binary to L-ary (and L-ary to binary) pre(post)conversion is required. Moreover, practice signal values are generally represented by finite number of bits, which can afford only a limited range of sample values. In certain cases, the embedding procedure outlined above may generate out-of-range sample values. For instance, in a 8 bpp representation (range is [0, 255]) the embedding algorithm with operating parameters L =6, $Q_L(s)$ =252, and $w = 5$ will output $s_w = 257$, which cannot be represented by an 8-bit

value. In general, for a given signal value, watermark symbols can only take N values (w is an N-ary symbol) where $N \leq L$. The out-of-range sample values can be avoided by skipping samples where $N < L$, at the expense of the embedding capacity at those samples. Alternatively, the binary to L-ary conversion algorithm presented below achieves the same objective without sacrificing the embedding capacity. The algorithm is motivated by arithmetic coding [14] for equi-probable input symbols. We start by interpreting the binary input string $\mathbf{h}$ as the binary representation of a number H in the interval [0,1), i.e., $H = .h_0h_1h_2\ldots$ and $H \in [0,1)$. Furthermore, we let R initially represent this interval ([0,1)). For our description, we assume the signal is encoded with integer values between zero and s_{max}.

1) Given s and s_{max}, determine $Q_L(s)$ and the number of possible levels $N = min(L, s_{max} - Q_L(s))$.
2) Divide R into N equal subintervals, R_0 to R_{N-1}.
3) Select the subinterval that satisfies $H \in R_n$.
4) Next watermark symbol is $w = n$.
5) Set $R = R_n$ and go to step 1), for the next sample.

This conversion process is illustrated in Fig. 3. Note that the inverse conversion is performed by the dual of the above algorithm. In particular, watermark symbols, $\mathbf{w}$, are converted into a binary number H by successively partitioning the interval $R = [0,1)$. Number of partitions (active levels) N on a given signal sample s_w are obtained from $Q_L(s_w) = Q_L(s)$. Pseudocode for this process is presented below.

1) Given s_w and s_{max}, determine $Q_L(s_w)$ and the number of possible levels $N = min(L, s_{max} - Q_L(s_w))$.
2) Divide R into N equal subintervals, R_0 to R_{N-1}.
3) Set $R = R_w$, where $w = s_w - Q_L(s_w)$ is the current watermark symbol.
4) If there are remaining symbols, go to step 1).
5) Find shortest binary string $H \in R$.

B. Embedding Capacity and Distortion

In G-LSB embedding (1), each signal sample carries an L-ary watermark symbol w_i, which represents $log_2(L)$ bits of information. Therefore, the *embedding capacity* of the system is

$$C_{\mathrm{GLSB}} = log_2(L) \tag{4}$$

bits per sample (bps) (barring boundary effects due to overflow restrictions).

A closed form expression for the expected mean square and mean absolute error distortions may be obtained if we assume that: 1) data symbols $\mathbf{w}$ are equi-probable, which is reasonable if the input data is compressed and/or encrypted, as in many data-embedding applications; and 2) the residual signal representing the L lowest levels of the original host signal ($r = s - Q_L(s)$), is uniformly distributed, which is a reasonable approximation for natural imagery, especially for small L

$$D(MSE) = \frac{1}{L^2}\sum_{r=0}^{L-1}\sum_{w=0}^{L-1}(r-w)^2 = \frac{L^2-1}{6} \tag{5}$$

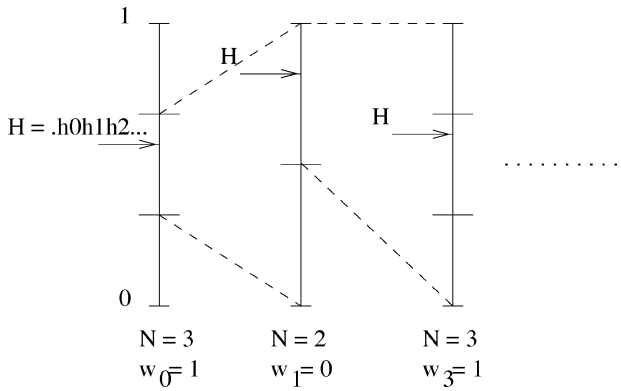

Fig. 3. Binary to L-ary conversion using a variant of arithmetic encoding.

$$D(MAE) = \frac{1}{L^2}\sum_{r=0}^{L-1}\sum_{w=0}^{L-1}|r-w| = \frac{L^2-1}{3L}. \tag{6}$$

III. Lossless Generalized-LSB Data Embedding

The G-LSB embedding algorithm outlined in the preceding section can be directly used for data embedding with low distortion. However, the method is irreversible, i.e., the host signal is permanently distorted when its lowest levels containing the residual signal are replaced with the watermark signal. This shortcoming can be remedied by including information for reconstruction of the residual signal along with the embedded data in the payload. This technique had been proposed in [12] and later used in [10], [11], [13] successfully.

Fig. 4 shows a block diagram of the proposed algorithm. In the embedding phase, the host signal $\mathbf{s}$ is quantized and the residual $\mathbf{r}$ is obtained (7). The residual is then compressed in order to create capacity for the payload data $\mathbf{h}$. The compressed residual and the payload data are concatenated and embedded into the host signal via G-LSB modification. In particular, the resulting bit stream is converted to L-ary symbols $\mathbf{w}$ and added to the quantized host to form the watermarked signal $\mathbf{s}_w$ (1). Note that the compression block uses the rest of the host signal, $Q_L(\mathbf{s})$, as side-information, in order to facilitate better compression and higher capacity.

In the extraction phase, the watermarked signal $\mathbf{s}_w$ is quantized and the watermark payload (the compressed residual and the payload data $\mathbf{h}$) is extracted (2). A desirable property of the proposed algorithm is that the payload data extraction is relatively simple, and it is independent of the recovery step. If desired, the algorithm proceeds with the reconstruction of the original host $\mathbf{s}$. In particular, the residual, $\mathbf{r}$, is decompressed using $Q_L(\mathbf{s}_w) = Q_L(\mathbf{s})$ as side-information. Original host, $\mathbf{s}$, is reconstructed by replacing the lowest levels of the watermarked signal with the residual (8)

$$\mathbf{r} = \mathbf{s} - Q_L(\mathbf{s}) \tag{7}$$

$$\mathbf{s} = Q_L(\mathbf{s}) + \mathbf{r} = Q_L(\mathbf{s}_w) + \mathbf{r}. \tag{8}$$

Note that the lossless-embedding system has significantly smaller capacity than the raw G-LSB scheme, since the compressed residual typically consumes a large part of the available

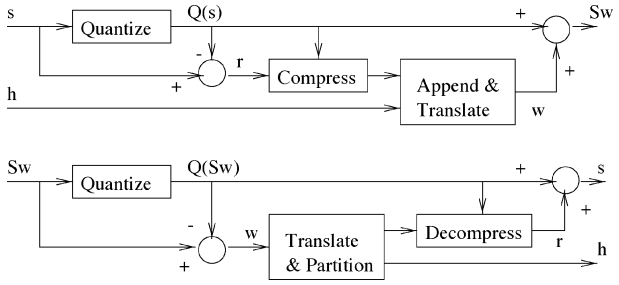

Fig. 4. (Top) Embedding and (bottom) extraction phases of the proposed lossless data-embedding algorithm.

capacity. The lossless-embedding capacity of the system is given by

$$C_{\text{Lossless}} = C_{\text{GLSB}} - C_{\text{Residual}}, \tag{9}$$

where C_{GLSB} is the raw capacity of G-LSB embedding (4) and C_{Residual} is the capacity consumed by the compressed residual. This observation emphasizes the importance of the residual compression algorithm; the better the compression, the higher the lossless-embedding capacity.

A. Compression of the Residual

Efficient compression of the residual is the key to obtaining high lossless-embedding capacity. Since the residual signal represents the lowest levels of a continuous-tone image (7), the compression is a challenging task. For small values of L, the residual typically has no structure, and its samples are virtually uniformly distributed and uncorrelated from sample to sample. Direct compression of the residual therefore results in a rather small lossless-embedding capacity. However, if the rest of the image information is used as side-information, significant coding gains can be achieved in the compression of the residual, by exploiting the spatial correlation among pixel values and the correlation between high and low levels (bit planes) of the image.

The proposed method adapts the CALIC lossless image compression algorithm [15], [16] for the lossless-embedding scenario. The algorithm is comprised of three main components: 1) prediction, 2) context modeling and quantization, 3) conditional entropy coding. The prediction component reduces spatial redundancy in the image. The context modeling stage further exploits spatial correlation and the correlation between different image levels. Finally, conditional entropy coding based on selected contexts translates these correlations into smaller code-lengths. The algorithm is presented below in pseudocode.

```
1) ŝ_O=Predict Current Pixel().
2) d, t=Determine Context D, T(ŝ_O).
3) ṡ_O=Refine Prediction (ŝ_O, d, t).
4) θ=Determine Context Θ(ṡ_O).
5) If (θ ≥ 0),
  Encode/Decode Residual (r_O, d, θ);
else,
  Encode/Decode Residual (L−1−r_O, d, |θ|).
```

1) Prediction: A local neighborhood of a pixel which consists of its eight-connected neighbors is seen in Fig. 5. In this neighborhood, we denote the current (center) pixel(residual) position by O, and neighboring positions by W, NW, N, NE, E, SE, S, and SW.[3] The residual samples are encoded and decoded in the raster scan order, i.e., left-to-right and top-to-bottom (Fig. 5). This order guarantees that residuals at positions W, NW, N, NE have already been reconstructed when the center residual, r_O, is being decoded. In addition, all quantized pixel values of the image, $Q_L(\mathbf{s})$, are known as side information. Therefore, at a given position, pixel values $s = Q_L(s) + r$ at positions W, NW, N, NE and quantized pixel values $Q_L(s)$ at positions E, SE, S, SW are known. To simplify the notation, we define a reconstruction function $f(.)$, which gives the best known value of a neighboring pixel, exact value if known, or the quantized value plus $(L/2)$ (to compensate for the bias in the truncation Q_L)

$$f(s_k) = \begin{cases} s_k, & \text{if } k \in \{W, NW, N, NE\}, \\ Q_L(s_k) + \frac{L}{2}, & \text{otherwise.} \end{cases} \tag{10}$$

A simple, linear prediction for the current pixel value is calculated using the nearest four-connected neighbors of a pixel

$$\hat{s}_O = \frac{1}{4} \sum_{k \in \{W,N,E,S\}} f(s_k). \tag{11}$$

Since this predictor is often biased, resulting in a nonzero mean for the prediction error $s_O - \hat{s}_O$, we refine this prediction and remove its bias using a feed-back loop, on a per-context basis as in [16]. The refined prediction is calculated as

$$\dot{s}_O = round\left(\hat{s}_O + \bar{\epsilon}(d,t)\right) \tag{12}$$

where $round()$ is the integer round and $\bar{\epsilon}(d,t)$ is the average of the prediction error $(\epsilon = s_O - \dot{s}_O)$ over all previous pixels in the given context (d,t). In order to avoid the propagation of rounding errors, the average prediction error is computed from the refined prediction instead of the raw prediction in (11). The resulting predictor $\dot{s}_O$ is a context-based, adaptive, nonlinear predictor [16].

2) Context Modeling and Quantization: Typical natural images exhibit nonstationary characteristics with varying statistics in different regions. This causes significant degradation in performance of compression algorithms that model the image pixels with a single statistical model such as a universal probability distribution. If the pixels can be partitioned into a set of *contexts*, such that within each context the statistics are fairly regular, the statistics of the individual contexts (e.g., probability distributions) may be exploited in encoding the corresponding pixels (residuals) using conditional entropy coding. If the contexts and the corresponding statistical models are chosen appropriately, this process can yield significant improvements in coding efficiency. The context selection problem addresses the fundamental trade-off concerning the number of contexts. Increasing number of contexts better adapt to the local image statistics hence improve the coding efficiency.

[3]The O indicates origin, and the symbols for other immediate neighbors are based on the directions in standard map orientation, wherein N denotes north, NW north west, and so on.

Since the corresponding conditional statistics often have to be learned on-the-fly observing the previously encoded (decoded) symbols, convergence of these statistics and, thereby, efficient compression is delayed when a large number contexts are used. The reduction in compression efficiency due to large number of contexts is known as the *context dilution* problem. A good context model should avoid context-dilution by choosing the optimum number of contexts.

As a first step, we adopt a variant of d and t contexts from [16], which are defined as follows:

$$\Delta = \sum_{k\in\{W,NW,N,NE,E,SE,S,SW\}} \frac{1}{8}\left|f(s_k) - \hat{s}_O\right| \quad (13)$$

$$d = Q(\Delta) \quad (14)$$

$$t_k = \begin{cases} 1, & \text{if } f(s_k) > \hat{s}_O \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

$$t = t_W||t_N||t_E||t_S \quad (16)$$

where t is obtained by concatenating the individual t_k bits (16 values), and $Q(\Delta)$ is a scalar nonuniform quantizer with eight levels, whose thresholds are experimentally determined so as to include an approximately equal number of pixels in each bin.[4] The context d corresponds to local activity as measured by the mean absolute error of the unrefined predictor (11) and t corresponds to a texture context that is based on the relations of the individual neighbors to the unrefined prediction.[5]

As described earlier in III-A-1, for each pixel, the (d, t) context is determined and the prediction is refined by using the average prediction error for the previous pixels in the context, as in (12). In the encoding step, the average prediction error for the context is then updated using the prediction error for the current pixel, in the decoding step, the pixel is first decoded and the update follows.

Typically, the probability distribution of the prediction error $\epsilon = s - \dot{s}$ can be approximated fairly well by a Laplacian distribution with zero mean and a small variance which is correlated with the context d [17, p. 33]–[19]. In order to make precise statements, for the following discussion, we assume that the prediction error distribution $p(\epsilon|d)$ is exactly Laplacian with variance σ_d determined by d. The arguments and the ensuing conclusions and techniques, however, are largely applicable even when the true distributions deviate from this assumption. Fig. 6(a) shows a plot of the probability mass function (pmf) $p(\epsilon|d)$ under this assumption. Given $\dot{s}$, the conditional probability distribution of pixel values $p(s = \dot{s} + \epsilon|d, \dot{s})$ is obtained by shifting the prediction error distribution $p(\epsilon|d)$ by $\dot{s}$. The corresponding pmf is illustrated in Fig. 6(b).

In typical lossless image compression applications, pixel values are coded using these conditional probability distributions. However, the residual compression problem set-forth in this paper deviates from the usual lossless image compression problem in two aspects. 1) The range of residuals, i.e., $[0, L)$, is typically much smaller than the range of pixel values, e.g., [0,255]. Therefore, instead of coding the complete range of pixel values, it is more efficient to encode only the residual. The residual's probability distribution for entropy encoding can be obtained from the pixel statistics. 2) The quantized value of the pixel $Q_L(s)$ is known, and this knowledge can be exploited. We address these issues by introducing an additional context θ, which is used only in the coding process and not in prediction.

[4]For the experimental results of Section IV, the quantizer $Q()$'s threshold are $\{1, 2, 3, 4, 6, 10, 15\}$.

[5]In order to avoid context-dilution during coding, t contexts are used only during prediction and not while coding.

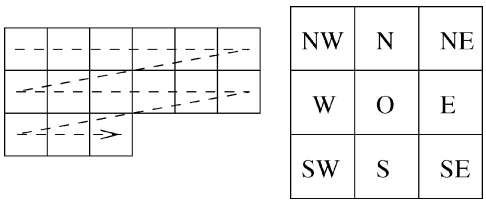

Fig. 5. (Left) Raster scan order and (right) an eight-connected neighborhood. Current pixel is in the center (O). Surrounding pixel positions are denoted by their relative directions, W, NW, N, NE, E, SE, S, and SW.

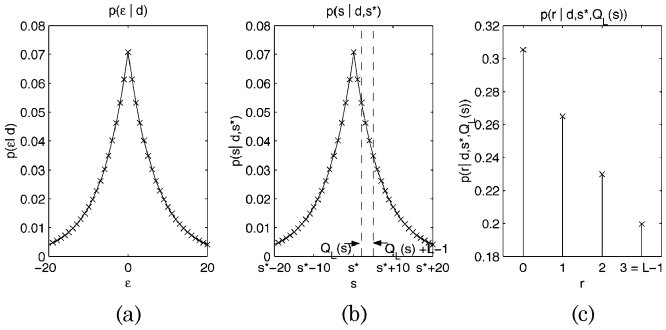

Fig. 6. (a) Prediction error PMF $p(\epsilon|d)$ under Laplacian assumption ($\sigma_d = 10$). (b) Corresponding pixel PMF $p(s = \dot{s} + \epsilon|d, \dot{s})$. (c) Conditional PMF of the residual (L =4), $p(r|d, \dot{s}, Q_L(s))$.

In order to motivate the context θ, note that the known quantized value $Q_L(s)$ may be used as an additional context directly. A known quantized pixel value $Q_L(s)$ limits the possible values of the pixel s to the range $[Q_L(s), Q_L(s) + L)$. This is illustrated in Fig. 6(b) as the region between the two vertical broken lines. The conditional probability mass function $p(r|d, \dot{s}, Q_L(s))$ can, therefore, be obtained by normalizing this segment of the probability mass function to sum up to 1. Fig. 6(c) illustrates the conditional probability mass function $p(r|d, \dot{s}, Q_L(s))$ obtained for the segment illustrated in Fig. 6(b). Entropy coding the residual using this conditional pmf restricts the symbol set required, thereby improving compression. Note, however, that there are typically a large number of possible values for $Q_L(s)$, which would cause significant context dilution since a large number of samples would be required to learn the statistics for each of these contexts on the fly. The characteristics of the Laplacian distribution, however, allow for a significant reduction in the number of these contexts.

Since the Laplacian distribution decreases exponentially about its peak at $\dot{s}$, the conditional pmf $p(r|d, \dot{s}, Q_L(s))$ can be determined from the relative positions of $\dot{s}$ and $Q_L(s)$. For instance, if $\dot{s} \leq Q_L(s)$, the peak is at $r = 0$ and the pmf decreases exponentially and is identical for all cases corresponding to $\dot{s} \leq Q_L(s)$. This case corresponds to the one illustrated in Fig. 6(b) and (c). This allows all the cases corresponding to $\dot{s} \leq Q_L(s)$ to be combined into a single composite context. Similarly, if $\dot{s} \geq Q_L(s) + L - 1$, the peak is at $r = L - 1$

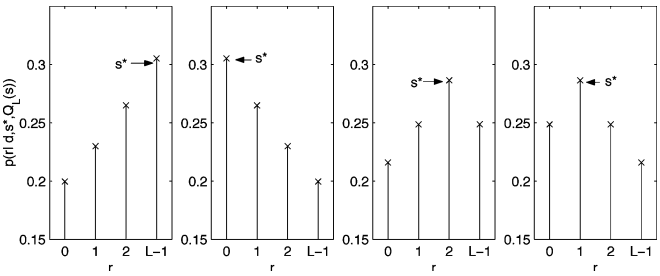

Fig. 7. Conditional PMFs $p(r|d, \dot{s}, Q_L(s))$ for contexts $\theta = \{\pm 1, \pm 2\}$ $(L = 4)$. Symmetric contexts are merged by remapping the residual values.

and the distribution increases exponentially, which may all be combined into a single context as well. In other cases, when $Q_L(s) < \dot{s} < Q_L(s) + L - 1$, the peak is at $r = \dot{s} - Q_L(\dot{s})$. Although total number of contexts after the above reductions is not large, it can be reduced further, if the symmetry of the Laplacian is exploited.

The symmetry of possible residual statistics is illustrated in Fig. 7. In particular, the distributions with peaks at r_θ and $L - 1 - r_\theta$ are mirror images of each other. If the residual values are remapped (flipped $r_{new} = L - 1 - r_{old}$) in one of these two contexts, the resulting distributions will be identical. As a result, we can merge these contexts without incurring any penalty. Furthermore, we encode the remapping instruction into the sign of the θ context. We assign each pair of symmetric distributions to an opposite sign, equal magnitude context value $(\pm\theta_i)$. During entropy encoding, first the residuals are remapped if necessary. Subsequently, the absolute value of θ is used as the coding context, together with d.

The θ contexts differentiate between statistically different (after incorporating all symmetries) residuals using the knowledge of $\dot{s}$ and $Q_L(s)$. This enables the conditional entropy coder to adapt to the corresponding probability distributions in order to achieve higher compression efficiency. Minimizing the number of such contexts allows the estimated conditional probabilities to converge to the underlying statistics faster. Therefore, it prevents context dilution and improves the compression efficiency.

In our experiments, we have observed that separating the case $\dot{s} = Q_L(s)$ from $\dot{s} < Q_L(s)$ and $\dot{s} > Q_L(s) + L - 1$ produces even better compression results. We believe that the rounding in (12) partially randomizes the distributions when $Q_L(s) \approx \dot{s}$ and causes this phenomenon. When the corresponding new contexts are created, total number of θ contexts equals $\lfloor (L + 1/2) + 1 \rfloor$. The total number of coding contexts (d, θ) is $8\lfloor (L + 1/2) + 1 \rfloor$.

3) Conditional Entropy Coding: In the final step, residual values are entropy coded using estimated probabilities conditioned on different contexts. In order to improve efficiency, we use a context-dependent adaptive arithmetic coder [14] as in [16]. In a context-dependent adaptive entropy coder, conditional probability distribution of residuals in each coding context (d, θ) is estimated from previously encoded(decoded) residual values. That is, the observed frequency of each residual value in a given context approximates its relative probability of occurrence. These frequency counts are passed to an arithmetic coder which allocates best code-lengths corresponding to given symbol probabilities.

B. Selective Embedding and Compression

In order to maximize lossless data-embedding capacity for a given embedding level, the algorithm presented above utilizes every pixel in a given image, i.e., the residual for each pixel is replaced and incorporated in the embedded data in compressed form and the lowest L signal levels of the pixel are used for embedding data. When a pixel is used for data embedding, it increases the expected embedding distortion by a fixed amount (5). The additional lossless-embedding capacity created by the pixel depends on the compressibility of the residual for the pixel, which in turn is may be approximated by the average codeword length for the corresponding context. The average codeword length varies significantly between different coding contexts. In our experiments, we have observed that residual compression is more effective in the "smooth" regions of an image, due to more accurate prediction of pixel values. This observation is supported by the steeper conditional residual probability mass functions (small variance) in contexts corresponding to small values of d, which roughly corresponds to the smooth regions of the image. As a result, using pixels in these contexts (regions) yields a higher embedding capacity for a fixed-embedding distortion. Conversely, pixels corresponding to contexts with large values of d contribute small or negative amounts of capacity while still contributing similar amounts of distortion. If these pixels were left unaltered and not included in the embedding process, one would obtain significant reduction in distortion without an appreciable capacity penalty.

An algorithm which utilizes a subset of all pixels (a subset mask) is called a *selective embedding* algorithm. In general, the mask that maximizes the capacity at a given embedding distortion may be found through an exhaustive search. Not only does this approach significantly increase the computational complexity, but also it requires a side channel for the transmission of the resulting mask (the mask determines which pixels carry the embedded payload; therefore, it should be available during extraction of the embedded payload). The algorithm we adopt here calculates a suboptimal mask for a given embedding distortion and level from either the original or watermarked host signal. The proposed algorithm utilizes the structured distortion property of the G-LSB embedding algorithm. It uses the quantized images $Q_L(\mathbf{s}) = Q_L(\mathbf{s}_w)$ as a common basis for mask calculation and, thus, avoids any reference to the residual. The ranking of conditional codeword lengths of each residual is estimated using the smoothness of the quantized signal at that position. In particular, for each pixel, s, the local variance in its four-connected quantized neighborhood is calculated

$$\sigma^2 = \frac{1}{4} \sum_{k \in \{W, NW, N, NE\}} (Q_L(s_k) - \mu)^2 \qquad (17)$$

where $\mu = (1/4) \sum_{k \in \{W, NW, N, NE\}} Q_L(s_k)$ is the mean of same four pixels. In our experiments, this smoothness measure is observed to be well correlated with the average codeword length: Codewords are shorter in the average when the variance is low. Later, starting from the pixel with the lowest local

variance, specified number of pixels are assigned to the mask. This assignment process can be simplified by collecting a histogram of variance values and determining a threshold where given number of pixels have a lower variance. Then, each pixel is assigned to the mask if its variance is less than the threshold. In practice, a discrete histogram of variance values is collected. Discrete valued histogram bins may lead to a threshold value which assigns more pixels to the mask than the number specified by the distortion. In this case, the assignment procedure is carried out in two passes. In the first pass, all pixels below the threshold except the ones which belong to the histogram bin immediately before the threshold are assigned to the mask. In the second pass, excluded pixels are traversed in a particular scan order and assigned to the mask until total number of pixels in the mask matches the specified number. Since this process uses the quantized images which are not modified by the data embedding, i.e., $Q_L(\mathbf{s}) = Q_L(\mathbf{s}_w)$, for a given embedding distortion, the results of the mask selection are identical for the embedding and extraction processes ensuring that synchronism is maintained.

This extension improves the capacity-distortion performance of the original algorithm, at the expense of increased computational burden of calculating the mask.

C. Capacity Control

Thus far, we have concentrated on maximizing the lossless-embedding capacity given an embedding level and a target distortion. In most practical applications, however, the complementary problem is of interest, where the goal is to determine the embedding level and mask which result in the least possible distortion, while providing the given target capacity. As the capacity consumed by the compressed residuals—thus, the lossless-embedding capacity—varies with the underlying image statistics, it is not possible to accurately estimate the embedding level and distortion from the target capacity.

A more suitable approach that is utilized here, is to vary the embedding level L and the target distortion in the method of Section III-B so as to achieve the desired capacity. In order to allow extraction and recovery, the embedding level L, and the final target distortion must be communicated to the receiver. This is achieved by modifying LSBs of a small number of pixels at fixed positions, say first n pixels, which are excluded during the lossless-embedding phase. In order to ensure recovery, the original LSBs of these pixels are transmitted as a part of the payload. In order to limit the required through-put from this additional channel, only a limited set of level-distortion pairs are used. After obtaining the level and distortion parameters through the additional fixed channel, the extraction algorithm calculates the embedding mask which maximizes the embedding capacity as outlined in Section III-B. As a result, the capacity-control problem, i.e., minimizing the distortion given a target capacity at the embedding phase, is reduced to finding the embedding level-distortion pair (from a limited set of such pairs) which has the minimum distortion and satisfies the capacity requirement. The achievable capacity of a given level-distortion pair is obtained by compressing the residual and calculating the G-LSB embedding capacity with the given parameters. An iterative algorithm, which searches through level-distortion pairs in the given limited set, finds the best operating point among all possible points. In a simple instance of such an iterative technique, the level-distortion pairs (possible operating points) are ordered starting from the lowest distortion. The algorithm sequentially computes the capacities for each element in this list, until the target capacity is met. If the initial target cannot be achieved under desired distortion constraints, some applications may alternatively allow for reduced target rates. For instance, the image authentication method in [7] can lower the required embedding rate at the expense of reduced tamper-localization accuracy.

Fig. 8. 512×512 grayscale images used for testing.

IV. Implementation and Experimental Results

The lossless G-LSB embedding algorithm and its selective embedding extension have been implemented and tested on a number of images. The images used in the evaluation are shown in Fig. 8. The images range from fairly smooth images, e.g., *F-16*, to highly textured images, e.g., *Mandrill*. We first present an overview of the implementation issues. Thereafter, the performance of the core algorithm and its extension are presented and compared to the existing schemes.

A. Implementation Issues

The adaptive binary to L-ary conversion algorithm, outlined in Section II-A, is a variant of the arithmetic (de)coding process with equal symbol probabilities. Therefore, an integer implementation of arithmetic coding from [14] is employed. Over the images listed, this implementation achieves a practical capacity within 4 bits of the theoretically estimated embedding capacity in (4). The same algorithm is also used as the adaptive arithmetic coder for residual compression.

In capacity-controlled selective embedding (Section III-C), level-distortion pairs are coded by the level and pixel-percentage pairs. In order to accelerate the optimization search and reduce the signaling overhead, possible embedding levels are limited to $L = [2, 16]$ and percentage of modified pixels is quantized to one of $\{25\%, 50\%, 75\%, 100\%\}$. These level-percentage pairs are further coded and represented by a single overhead byte. The values mentioned here are selected for illustrative purposes and can be customized based on the individual requirements of specific applications.

In order to guarantee flawless operation, we define a simple protocol and an associated syntax for the embedded payload. First, the overhead byte representing the embedding parameters is embedded in the LSBs of first eight pixels of the raster scan order. The original values of these pixel LSBs are buffered and these positions are excluded in any subsequent embedding mask. Payload to be embedded using the G-LSB algorithm consists of four parts: 1) length of message data in bytes, 2) message data, 3) buffered LSBs, and 4) compressed residual description. The length of the message data is represented by two bytes allowing for a maximum length of 64 K bytes (a variable size length descriptor can be utilized if message data is expected to be larger). The length of the compressed description is not specified since the total number of coded symbols (residuals) is defined by the pixel-percentage parameter. The payload data constructed according to this syntax is embedded into the image using G-LSB embedding algorithm. A total of three bytes, two-byte length, plus the overhead byte, is the overall overhead. In applications where a fixed-length message is embedded, the overhead can be reduced accordingly.

B. Experimental Results

The lossless-embedding capacity-distortion performance of the proposed algorithm has been evaluated for the 512×512 grayscale images seen in Fig. 8. The lossless capacity obtained by embedding at various levels for each of these six images is plotted against the embedding level L in Fig. 10. A subset of these results is tabulated in Table I, where the average embedding distortion induced when the full capacity is utilized is also included. These average distortion values obtained from the watermarked images agree quite well with the theoretical results from (5). From Fig. 10 and Table I, we can see that the capacity of the proposed method is heavily dependent on the characteristics of the host image. Images with large smooth regions, e.g., *F-16*, accommodate higher capacities than images with irregular textures, e.g., *Mandrill*. In smooth regions, the predictor is more accurate, and, therefore, conditional residual distributions are steeper with smaller variances. These distributions result in shorter code lengths and, thus, higher embedding capacities.

For each individual image, the capacity of the scheme increases roughly linearly with number of levels (or exponentially with number of bit planes). This is due to stronger correlation among more significant levels (bit planes) of the image and the algorithm's ability to exploit this correlation. The rate of the increase, however, is not entirely constant either among images or throughout the levels.

The visual impact of the data embedding can be seen in Fig. 9, where the original "Gold" image is compared to the watermarked version in which a random message of over 3400 Bytes (27200 bits) is embedded using an embedding level of $L = 6$. The visual distortion introduced by the embedding is quite small, though, upon close inspection, a small amplitude white noise may be observed in the watermarked version. Common applications of lossless embedding such as meta-data tagging and authentication have fairly small to modest capacity requirements, which can be met with relatively small visual distortion in the watermarked image.

TABLE I
AVAILABLE CAPACITY (BYTES) VERSUS EMBEDDING LEVEL AND AVERAGE PSNR (DECIBELS) FOR EACH TEST IMAGE

Level(L)	2	3	4	5	6	8	10	12	14	16
PSNR(dB)	51.1	46.9	44.2	42.1	40.5	38.0	36.0	34.4	33.0	31.9
F-16	2190	4757	7640	10155	13452	17863	22673	26962	30828	34239
Mandrill	80	241	458	746	1112	1910	2815	3849	4640	5793
Boat	629	1707	3074	4620	6219	9835	13190	16313	18674	22282
Barbara	558	1503	2692	4094	5551	8298	11198	13622	16172	17604
Gold	309	879	1576	2451	3441	5656	8007	10445	12376	14569
Lena	598	1537	2856	4297	5893	9347	12712	15768	19117	22127

Fig. 9. (Left) Original and (right) watermarked *Gold* image. (PSNR $= 40.5$ dB, 3441 bytes embedded, and embedding level $\mathrm{L} = 6$).

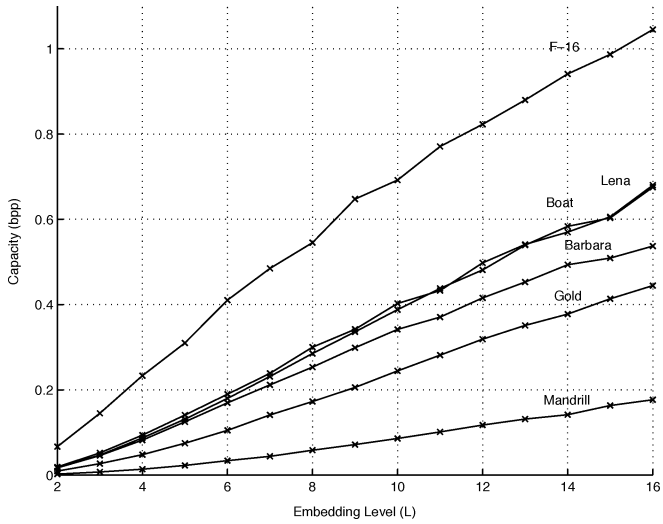

Fig. 10. Lossless embedding capacity C_{Lossless} versus embedding level L for all images.

The images in the test set above were all of a fixed size of 512×512 pixels. In order to test the effects of image size on the embedding capacity, the *Gold* image was divided into four quadrants and the subimages corresponding to each quadrant was independently used for lossless embedding. At level $L = 2$, the capacity degrades by 13% as result of subdividing the image. Starting from top-left and in clockwise direction embedding capacities are 83, 94, 61, and 30 bytes with a total of 268 bytes instead of 309 bytes. This reduction is induced by the modeling cost associated with multiple contexts: the adaptive coding scheme employed here requires a learning stage—when the algorithm adapts to the particular image's statistics—during which the compression efficiency is low. For smaller images,

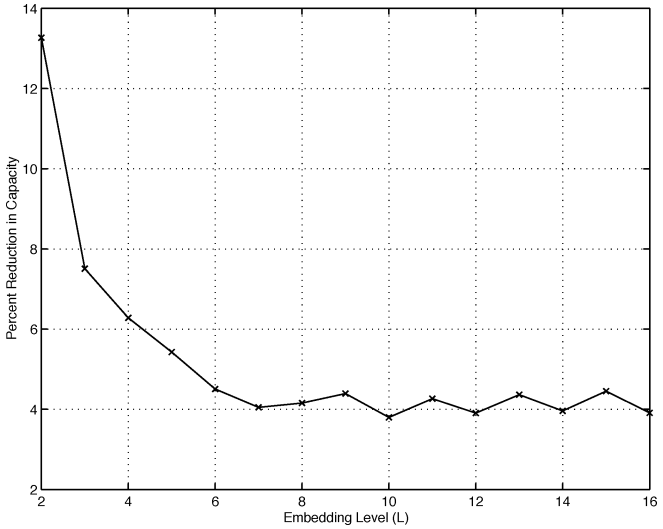

Fig. 11. Percent capacity degradation due to independent processing of quadrants of *Gold* image as a function of embedding level L.

TABLE II
SELECTIVE EMBEDDING EXTENSION: AVAILABLE CAPACITY (BYTES) AND AVERAGE PSNR (DECIBELS)

Level	% Pixels used	PSNR (dB)	F-16	Mandrill	Barbara
2	25	57.1	1393	50	302
2	50	54.1	1906	73	505
2	75	52.4	2159	76	553
2	100	51.1	2190	80	558
4	25	50.2	2666	301	1410
4	50	47.2	5731	424	2413
4	75	45.5	7452	445	2668
4	100	44.2	7640	458	2692
8	25	44.0	4643	1181	2965
8	50	41.0	10964	1721	7233
8	75	39.4	17010	1850	8199
8	100	38.0	17863	1910	8298

the relative overhead of the learning stage is larger because it is amortized over fewer pixels. Similarly, the signaling overhead has a bigger impact in smaller images.

Fig. 11 shows the capacity degradation—as a percentage of the initial capacity—on the *Gold* image for embedding levels $L = 2, 3, \ldots 16$. In all cases, the subdivision into smaller images reduces the capacity but the percentage reduction in capacity shows a predominantly decreasing trend with increasing embedding level L. At a first glance, this observation contradicts the modeling cost explanation: The number of coding contexts (d, θ) increases with increasing embedding level. One would, therefore, expect a correspondingly greater loss in the smaller subdivided images due to the increase in adaptation overhead of the context adaptive coder to these individual contexts. This explanation, however, implicitly assumes that the residual statistics at each embedding level are similar, and, therefore, the adaption overhead is directly related to the number of contexts. In practice, the residual statistics differ significantly for different embedding levels. At higher embedding levels, the stronger correlation between image levels results in a distribution of the residuals with a smaller variance in comparison to the energy of the residual. As a result, reasonable approximations to residual statistics at these levels can be obtained more quickly, i.e., the adapting requires a smaller number of samples. Consequently, the relative impact of the adaptation overhead with smaller subdivided images is reduced as the adaptation level L increases, even though the number of coding contexts increases. The contribution of the fixed signaling overhead is also de-emphasized at higher levels, due to the net increase in overall capacity.

The three images, *Mandrill*, *Barbara*, *F-16*, corresponding, respectively, to low, medium- and high-embedding capacities were selected as representatives for the evaluation of the selective embedding extension to the lossless embedding algorithm described in Section III-B. The lossless embedding capacity and distortion values obtained for embedding levels $L = 2$, 4, and 8 by utilizing one to four fourths, i.e., 25%, 50%, 75%, and 100% of all pixels in data embedding for each of these cases is listed in Table II. The specific pixels corresponding to these individual percentages are appropriately chosen so as to minimize distortion using the process described in Section III-B. Note that these percentage points are chosen for illustrative purposes. A different or a larger set of percentage values may be used, if desired. Likewise, other embedding levels can also be utilized. In Table II, we observe that the minimum achievable distortion is reduced (down to 57.1 dB at $L = 2$ with 25% pixels selected for embedding) by the extension. This property is useful in applications which require small capacities and ultralow embedding distortions. The selective embedding extension also offers additional operating points with different distortion values which further improves the capacity distortion scalability provided by the different embedding levels of the G-LSB algorithm. Since the selective embedding process chooses pixels that are likely to provide the largest embedding capacity and, therefore, its capacity does not drop in proportion to the percentage of pixels used in embedding but at a much smaller rate. This is apparent in Table II where we see that the embedding distortion may be reduced by at least 1 dB with a negligible (less than 2%) reduction in the embedding capacity. The selective modification extension therefore leads to improved capacity-distortion performance. For instance, the embedding capacity for *Mandrill* at approximately 47 dB is 241 Bytes in the original method (Table I). On the other hand, the extension yields a capacity of 424 Bytes—a 76% increase—at a slightly lower distortion (47.2 versus 46.9 dB in the original algorithm) when 50% of all pixels are used. Similar trends are seen at various points for other images, as well.

The capacity-distortion performance of the baseline lossless GLSB algorithm (100%) and the selective embedding extension at the chosen embedding percentages of 75%, 50%, and 25% is compared for the *Barbara* image in Fig. 12, where the lossless-embedding capacity is plotted against the PSNR for different values of the embedding level L for each of these embedding percentages. As expected, a reduction in embedding percentage reduces the lowest achievable distortion (corresponding

to $L = 2$) pushing the lower ends of the corresponding curves further to the right. In most cases, little or no capacity penalty is incurred in the process for this lowest point. As a result, at the low end of the distortion scale, reducing the embedding percentage causes the capacity-distortion curve to move to the right, adding lower distortion points or improving capacity distortion performance in this region. Reducing the embedding percentage does not, however, guarantee a universally superior capacity distortion performance. This is seen in Fig. 12 for the *Barbara* image, where the performance of 25% selective embedding extension drops below the original (100%) scheme as one moves to the higher distortion regions (higher values of L). The deterioration in capacity-distortion performance can be attributed to the reduced effectiveness of the selection process at higher embedding levels. As indicated in Section III-B, to ensure the synchronization between the embedder and the detector, the selection of embedding regions is based on the smoothness of the quantized images $Q_L(\mathbf{s})$, and is therefore progressively impaired as the embedding level increases and correspondingly the quantization noise. Thus the simple selection criterion based on (17) can cause in exclusion of pixels whose inclusion in the embedding would be beneficial from a capacity-distortion perspective. In addition, the reduced number of pixels in the selective embedding also increases the impact of the modeling cost. The combined effect of these factors can result in poorer capacity-distortion performance for the selective embedding at higher distortions.

C. Performance Comparison With Prior Art

We compare the performance of the proposed method against other Type-II lossless data-embedding algorithms. Type-I algorithms are excluded from this comparison due to their inferior distortion-capacity performances.

1) Independent Bit-Plane Compression Methods: Early work [12], [13] on Type-II lossless embedding proposed compressing one of the LSB planes of the image by either using a bi-level lossless image compression algorithm, e.g., JBIG, or by using a dictionary based general purpose compression scheme, e.g., LZW (*gzip*). We replicated these methods by extracting the different LSB-planes of the images and compressing them by JBIG and *gzip* compression utilities. The capacity estimates[6] obtained through this process are tabulated in Table III. Note that the bit planes are numbered starting from the least significant, #1, proceeding with more significant ones (LSB #4 corresponds to the fourth least significant bit plane). From the table, it is apparent that direct compression approaches which attempt to compress the residual signal—or LSB planes—alone without utilizing the rest of the image perform significantly worse than the proposed scheme. In many cases, the compression algorithms actually expand the data. The conditional entropy coding scheme adopted here, however, successfully exploits the intra pixel correlation among the different levels of the same pixel and the inter-pixel correlations among neighbors, thus provides improved embedding capacities even at low distortions.

[6]In these experiments, file sizes are compared; it may be possible to slightly increase these capacity values by eliminating parts of the file headers, e.g., image size.

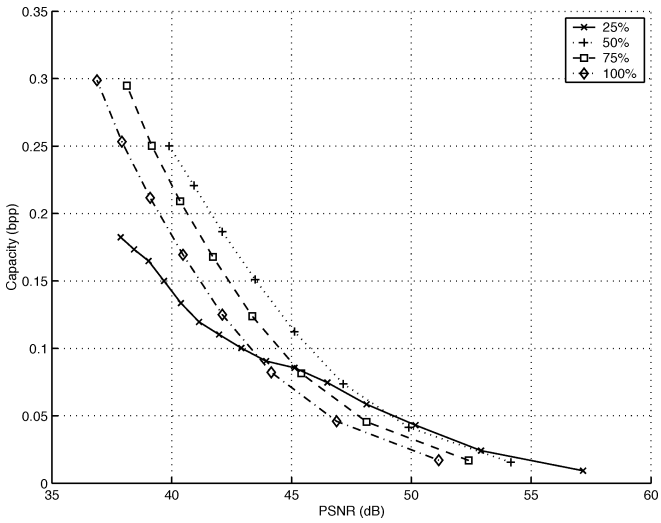

Fig. 12. Capacity-distortion performance of selective embedding extension on *Barbara* image.

2) RS Embedding Method: Recently, Goljan *et al.* [10] reported an elegant Type-II lossless data-embedding method called *RS Embedding* which offers significantly better results. In this method, first a flipping function $F_A(\cdot)$ is defined. The flipping function $F_A(\cdot)$ is a permutation on the set of possible pixel values, which swaps pairs of pixel values, separated by a defined amplitude A. $F_1(\cdot)$ with amplitude of one corresponds to LSB flipping. Small groups of pixels from the image, Gs, are classified into regular (R), singular (S), or unusable (U) classes based on the flipping function and a suitably chosen discriminant function, $f(\cdot)$. In particular, a group G is regular if $f(F_A(G)) > f(G)$, singular if $f(F_A(G)) < f(G)$ and unusable otherwise, where the flipping function is applied to one or more the pixels constituting the group. The discriminant function is intended to measure the smoothness of the group of pixels and for suitably chosen pixel groups and discriminant function, a majority of pixel groups appear regular in typical natural images [10].

From the definition of R, S, and U groups, it is apparent that the flipping function transforms an R group of pixels in to an S group, and S group into an R group and a U group in to a (different) U group. In addition, application of the flipping function twice to any group restores it to its original pixel values. These properties are the basis for information embedding and recovery in the *RS embedding* method. A binary payload is embedded in an image as a string of R and S features where, for instance, R represents the binary value 1 and S represents 0. U groups are skipped in the embedding and extraction process. The image pixel groups are scanned in a predefined order. For each R or S group, if the R/S classification matches the payload bit to be embedded, it is left unaltered; if not, the flipping function is applied to the group to ensure that the R/S classification matches the payload bit. To enable recovery of the original host at the receiver, the bit stream of RS features corresponding to the original image is compressed and included in the embedded payload. Thus, in the embedding, first the pixel groups in the original host image are scanned in a predefined order and the status of R and S groups is computed as a bit stream ($R = 1$ and

TABLE III
AVAILABLE CAPACITY (BYTES) AND AVERAGE PSNR (DECIBELS) WHEN ONE OF THE LSB-PLANES IS COMPRESSED WITH LZW OR JBIG ALGORITHMS. ZERO CAPACITY INDICATES NO OR NEGATIVE CAPACITY

Bit-plane	1	2	3	4
PSNR(dB)	51.1	45.1	39.1	33.1
F-16 (LZW)	0	7	1033	4756
F-16 (JBIG)	0	0	1264	7928
Mandrill (LZW)	0	0	0	0
Mandrill (JBIG)	0	0	0	0
Barbara (LZW)	0	0	0	1068
Barbara (JBIG)	0	0	0	1174

$S = 0$), with the U groups simply skipped in the process. The RS bit stream is then compressed to obtain a smaller bit-stream C. In the specific implementation used here, a binary adaptive arithmetic coder is used to compress the original string of R, S features. The message data is concatenated with the compressed bit-stream C to form the payload which is finally embedded in the image as outlined above. Suitable header information is included while concatenation to allow it to be undone.

In the data extraction and recovery process, the R and S bit-stream ($R = 1$ and $S = 0$) of the watermarked image is computed by scanning the image groups in the same order as the embedding process (once again ignoring U groups). The bit stream is partitioned in to the extracted message data and the compressed bit-stream C representing the R, S values for the original host image by reversing the concatenation step. Decompression of C yields the R, S values for original host. Finally, the original host is recovered by again scanning the image pixel groups and restoring the original R/S status of the groups by applying the flipping function to the R/S groups whose classification differs from the original.

As in all Type-II lossless-embedding schemes, the capacity available for lossless embedding depends on the compressibility of the string representing the original R, S features. The compression scheme exploits the imbalance between the number of R and S groups. The 0th order entropy of the R and S bit stream corresponding to the original host image, therefore, provides an accurate estimate of the fraction of the capacity that is consumed by the compressed recovery information, and equivalently the lossless-embedding capacity.

The capacity of the *RS Embedding* scheme depends on the specific choices for the pixel scan-order, the pixel groups G, the discriminant function $f(\cdot)$, and the amplitude A of the flipping function. Increasing amplitude A typically causes a monotonic increase in the lossless-embedding capacity and in the embedding distortion. For a given flipping amplitude A, the imbalance between the number of R and S groups is strongly dependent on the choice of the discriminant function and pixel groups. For our evaluation, we consider the two-pass "checkerboard" scheme that offers the highest capacity among the options evaluated in [10]. In this scheme, the image is divided into "Black" and "White" pixels in the same way as a chess board [the pixel at (i, j) is black if i + j is odd, and white otherwise]. In the first pass, the black pixels are traversed in raster scan order (left to right and top to bottom) and the white pixels are scanned in the second pass similarly. The pixel group for computing the discriminant function is defined as the nearest four-connected neighbors of the current pixel being traversed, with the origin O representing the current pixel and the four neighbors W, N, E, and S, as defined in Fig. 5. The discriminant function for a grouping centered at origin O defined as

$$f(s_O, s_W, s_N, s_E, s_W) = \sum_{k \in \{W,N,E,S\}} |s_O - s_k| \qquad (18)$$

and the application of flipping function to a group consists of flipping the value for the pixel at the origin using the defined amplitude flipping function leaving other pixels unaltered. Note that the pixel groups in this scheme are overlapping; however, the restriction of the flipping function to the pixel at the origin and use of the checkerboard pattern ensures proper embedding and recovery, since each pixel is traversed and potentially flipped only once.

TABLE IV
AVAILABLE CAPACITY (BYTES) AND AVERAGE PSNR (DECIBELS) CORRESPONDING TO SELECTED AMPLITUDES FOR RS VECTOR ALGORITHM

Amplitude A	1	2	3	4	5	6
PSNR(dB)	52.9	46.4	42.5	39.8	37.8	36.1
F-16	1529	4182	6604	8402	9746	10901
Mandrill	66	257	573	987	1439	1940
Barbara	327	1228	2434	3632	4818	5758

The version of *RS Embedding* outlined above was tested on the images, *Mandrill*, *Barbara*, and *F-16*. The amplitude for the flipping function A was varied from 1 to 6 in order to explore the capacity-distortion performance of the scheme. Table IV lists the capacity obtained (in bytes)[7] and average embedding distortions for each of these embedding amplitudes. In Fig. 13, capacity-distortion performance of the *RS Embedding* scheme is compared with the lossless G-LSB algorithm (at 100% embedding). From a capacity distortion perspective, the lossless GLSB (LGLSB) algorithm outperforms *RS Embedding* at most points with the exception of the lowest distortion points corresponding to $L = 2$ and $A = 1$. In this range, RS Embedding achieves an embedding capacity comparable to LGLSB with a significantly lower distortion.

Though *RS Embedding* with $A = 1$ and LGLSB with $L = 2$, both modify the LSBs of the pixel values similarly, the *RS Embedding* has a distortion advantage since it modifies only pixels corresponding to R and S groups, while skipping U groups. As a result, pixels belonging to U groups do not incur any embedding distortion, and a lower overall distortion is attained. The selective embedding extension allows LGLSB to similarly extend its domain of operation to lower distortions. In order to illustrate this, the selective embedding LGLSB at 75% is compared with *RS Embedding* in Fig. 14, where the capacity is plotted along

[7]These results are not adjusted for the signaling overhead (up to 3 bytes).

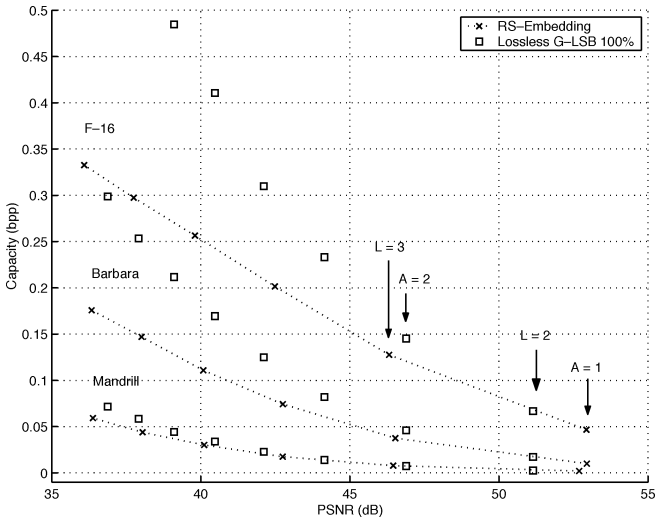

Fig. 13. Capacity-distortion performance comparison between baseline (100%) lossless GLSB-embedding algorithm and RS-embedding.

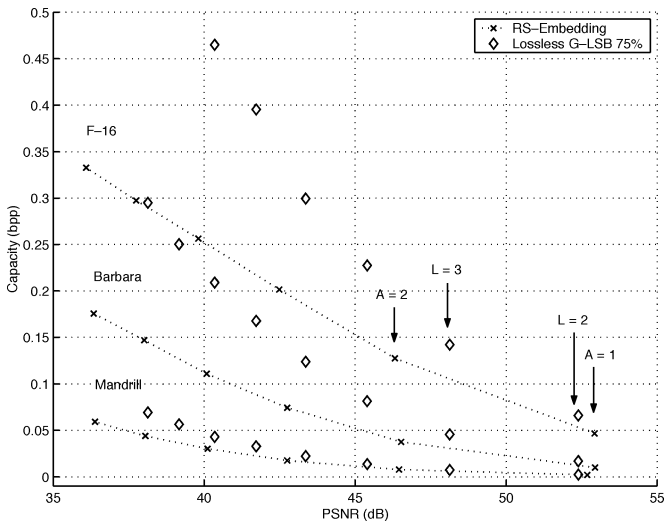

Fig. 14. Capacity-distortion performance comparison between extended lossless GLSB-embedding algorithm at 75% embedding and RS-embedding.

the ordinate with the embedding distortion along the abscissa for the two schemes. At $L = 2$ the LGLSB induces a distortion value similar to that of the *RS Embedding* at $A = 1$, with slightly higher capacity. For each of the three images, the LGLSB extension at 75% embedding capacity distortion curves lie above the corresponding curves for *RS Embedding*, indicating the improved performance.

The LGLSB algorithm and its selective embedding extension has an advantage over the *RS Embedding* scheme [10] in three respects.

1) Lossless G-LSB allow greater flexibility and finer capacity-distortion granularity. In particular, selective embedding scheme can modify an arbitrary percentage of pixels at a given level. Thus, it can operate at a specified distortion value. This property surpasses the granularity offered by different amplitudes in *RS Embedding*, unless the embedding amplitude is varied from pixel to pixel at the expense of additional complexity and signaling overhead.
2) The proposed method can achieve a significantly higher embedding capacity for a comparable distortion. This demonstrates the effectiveness of the selected compression technique. The capacity differential increases as the allowable embedding distortion is increased and higher levels or amplitudes are used. Besides the effectiveness of the compression, the higher capacity difference at higher embedding levels is related to the different pixel modification methods employed by each algorithm. In *RS Embedding* each pixel is modified by a single amplitude, A, to represent (encode) a binary symbol. This process has an irreversible embedding capacity of 1 bpp and induces an average MSE distortion of $A^2/2$. On the other hand, in G-LSB embedding each pixel is modified by a multitude of possible values, and it represents (encodes) an L-ary symbol. This induces an average MSE distortion of $L^2 - 1/6$ with an irreversible capacity of $log_2(L)$ bpp. Note that the lossless embedding capacity does not scale accordingly because L-ary symbols have longer associated codeword lengths during compression. Nevertheless, the compression scheme utilizes the redundancy among different image levels—and LL-ary symbol values—and as a result the lossless-embedding capacity is further improved.
3) The lossless G-LSB embedding can achieve capacities exceeding 1 bpp, while *RS Embedding*'s capacity performance is bounded from above by 1 bpp, regardless of the allowable distortion. When images with large very smooth regions are used, e.g., document images, the original image features can be compressed very effectively and the lossless-embedding capacity approaches the irreversible-embedding capacity values mentioned above. In these cases, G-LSB scheme may achieve capacities well above 1 bpp by increasing the embedding distortion.

Despite the above disadvantages, the *RS embedding* algorithm has significant complexity advantages over the LGLSB method. It has a relatively simpler implementation, lower memory requirements and lower computational costs.

3) Difference Expansion Method: A type-II embedding algorithm based on reversible integer transforms and *difference expansion* is proposed in [11]. The algorithm particularly emphasizes high payloads. For the *Lena* image the comparisons included in [11] demonstrate that for the higher capacity (and correspondingly higher distortion PSNR< 45 dB) the technique provides better embedding capacities. Low-distortion regions are not included in the comparisons.

V. Conclusion

A novel lossless (reversible) data embedding (hiding) technique is presented. The technique provides high-embedding capacities, allows complete recovery of the original host signal, and introduces only a small distortion between the host and image bearing the embedded data. The capacity of the scheme depends on the statistics of the host image. For typical images, the scheme offers adequate capacity to address most applications. In applications requiring high capacities, the scheme can

be modified to adjust the embedding parameters to meet the capacity requirements, thus trading off intermediate distortion for increased capacity. In such scenarios, the G-LSB embedding proposed in the current paper is significantly advantaged over conventional LSB-embedding techniques because it offers finer grain scalability along the capacity distortion curve. The performance of the algorithm—and its extensions—is rigorously tested with representative images and compared with the earlier methods. The proposed algorithm is shown to out-perform bit-plane compression and RS embedding methods, especially at moderate- to high-distortion regions.

REFERENCES

[1] R. L. Lagendijk, G. C. Langelaar, and I. Setyawan, "Watermarking digital image and video data," *IEEE Signal Process. Mag.*, vol. 17, no. 5, pp. 20–46, Sep. 2000.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, no. 7, pp. 1079–1107, Jul. 1999.

[3] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, no. 6, pp. 1064–1087, Jun. 1998.

[4] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "A hierarchical image authentication watermark with improved localization and security," in *Proc. IEEE ICIP*, Thessaloniki, Greece, Oct. 2001, pp. 502–505.

[5] ——, "Hierarchical watermarking for secure image authentication with localization," *IEEE Trans. Image Process.*, vol. 11, no. 6, Jun. 2002.

[6] National Electrical Manufacturers Association (NEMA), Digital Imaging and Communications in Medicine (DICOM), 2003.

[7] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Localized lossless authentication watermark (LAW)," *Proc. SPIE*, vol. 5020, no. 1, Jan. 2003.

[8] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," U.S. Patent #6 278 791, Aug. 2001.

[9] C. De Vleeschouwer, J. F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 97–105, Mar. 2003.

[10] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—new paradigm in digital watermarking," *EURASIP J. Appl. Sig. Process.*, vol. 2002, no. 02, pp. 185–196, Feb. 2002.

[11] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[12] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," *Proc. SPIE*, no. 1, pp. 197–208, Jan. 2001.

[13] J. Dittmann, M. Steinebach, and L. C. Ferri, "Watermarking protocols for authentication and ownership protection based on timestamps and holograms," *Proc. SPIE*, no. 1, pp. 240–251, Jan. 2002.

[14] I. H. Witten, M. Radford, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[15] X. Wu and N. Memon, "Context-based, adaptive, lossless image codec," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[16] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Trans. Image Process.*, vol. 6, no. 5, pp. 656–664, May 1997.

[17] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[18] J. B. O'Neal, "Predictive quantizing differential pulse code modulation for the transmission of television signals," *Bell Syst. Tech. J.*, no. 5, pp. 689–721, May 1966.

[19] J. B. Weinberger, J. B. Seroussi, and J. B. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

Mehmet Utku Celik (S'98) received the B.Sc. degree in electrical and electronic engineering from Bilkent University, Ankara, Turkey, in 1999 and the M.Sc. degree in electrical and computer engineering from the University of Rochester, Rochester, NY, in 2001, where he is currently pursuing the Ph.D. degree.

Currently, he is a Research Assistant in the Electrical and Computer Engineering Department, University of Rochester. His research interests include digital watermarking and data hiding—with emphasis on multimedia authentication—image and video processing, and cryptography.

Mr. Celik is a member of the ACM.

Gaurav Sharma (SM'00) received the B.E. degree in electronics and communication engineering from Indian Institute of Technology (formerly University of Roorkee), Roorkee, India in 1990, the M.E. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1992, and the M.S. degree in applied mathematics and the Ph.D. degree in electrical and computer engineering from North Carolina State University (NCSU), Raleigh, in 1995 and 1996, respectively.

From August 1992 through August 1996, he was a Research Assistant at the Center for Advanced Computing and Communications, Electrical and Computer Engineering Department, NCSU. From August 1996 through August 2003, he was with Xerox Research and Technology, Webster, NY, initially as a member of research staff and subsequently at the position of Principal Scientist. Since fall 2003, he has been an Associate Professor at the University of Rochester, Rochester, NY. His research interests include multimedia security and watermarking, color science and imaging, signal restoration, and halftoning.

Dr. Sharma is a member of Sigma Xi, Phi Kappa Phi, Pi Mu Epsilon, and IS&T. He was the 2003 chair for the Rochester chapter of the IEEE Signal Processing Society and currently serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and the SPIE/IS&T *Journal of Electronic Imaging*.

Ahmet Murat Tekalp (S'80–M'84–SM'91–F'03) received the M.S. and Ph.D. degrees in electrical, computer, and systems engineering from Rensselaer Polytechnic Institute (RPI), Troy, NY, in 1982 and 1984, respectively.

From December 1984 to August 1987, he was with Eastman Kodak Company, Rochester, NY. He joined the Electrical and Computer Engineering Department, University of Rochester, Rochester, in September 1987, where he is currently a Distinguished Professor. Since June 2001, he has also been with Koc University, Istanbul, Turkey. He has served as an Associate Editor for the Kluwer Journal *Multidimensional Systems and Signal Processing* (1994–2002). He was an area editor for the Academic Press Journal *Graphical Models and Image Processing* (1995–1998). He was also on the editorial board of the Academic Press Journal *Visual Communication and Image Representation* (1995–2002). At present, he is the Editor-in-Chief of the *EURASIP Journal on Image Communication* published by Elsevier. He is the author of *Digital Video Processing* (Englewood Cliffs, NJ: Prentice Hall, 1995). He holds five U.S. patents and his group contributed technology to the ISO/IEC MPEG-4 and MPEG-7 standards. His research interests are in the area of digital-image and video processing, including video compression and streaming, video filtering for high-resolution, video-segmentation, object-tracking, content-based video analysis and summarization, multicamera surveillance video processing, and the protection of digital content.

Prof. Tekalp was named as Distinguished Lecturer by IEEE Signal Processing Society in 1998. He has chaired the IEEE Signal Processing Society Technical Committee on Image and Multidimensional Signal Processing (January 1996 to December 1997). He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1990–1992) and the IEEE TRANSACTIONS ON IMAGE PROCESSING (1994–1996). He was appointed as the Technical Program Chair for the 1991 IEEE Signal Processing Society Workshop on Image and Multidimensional Signal Processing, the Special Sessions Chair for the 1995 IEEE International Conference on Image Processing, the Technical Program Co-Chair for IEEE ICASSP 2000, Istanbul, Turkey, and the General Chair of IEEE International Conference on Image Processing (ICIP), Rochester, in 2002. He is the Founder and first Chairman of the Rochester Chapter of the IEEE Signal Processing Society. He was elected as the Chair of the Rochester Section of IEEE for 1994 to 1995.

Eli Saber (S' 91-M'96-SM'00) received the B.S. degree in electrical and computer engineering from the University of Buffalo, Buffalo, NY, in 1988 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, Rochester, NY, in 1992 and 1996, respectively.

He joined Xerox Corporation, Webster, NY, in 1988 and is currently a Product Development Scientist and Manager leading the Image Science, Analysis, and Evaluation Area in the Print Engine Development Unit. He is an Adjunct Faculty Member at the Electrical and Computer Engineering Departments of the University of Rochester and the Rochester Institute of Technology, where he is responsible for teaching graduate coursework in signal, image, and video processing and performing research in digital libraries and image understanding. He has authored a number of conference and journal publications in the fields of signal, image, and video processing. He is also an Associate Editor for the *Journal of Electronic Imaging*. His research interests include color-image processing, image/video segmentation and annotation, content-based image/video analysis and retrieval, computer vision, and watermarking.

Dr. Saber is a member of the Electrical Engineering Honor Society, Eta Kappa Nu, the Imaging Science and Technology Society. He is also an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and he was Guest Editor for the special issue on color image processing for the IEEE SIGNAL PROCESSING MAGAZINE. He was appointed the Finance Chair for the IEEE International Conference on Image Processing (ICIP) 2002, Rochester, NY. He was also the General Chair for the Western New York Imaging Workshop in 1998. He was the recipient of the Gibran Khalil Gibran Scholarship and of several prizes and awards for outstanding academic achievements from 1984 to 1988, as well as the Quality Recognition Award in 1990 from The Document Company, Xerox. He