REVERSIBLE DATA HIDING

Mehmet U. Celik[a], *Gaurav Sharma*[b], *A. Murat Tekalp*[a,c], *Eli Saber*[b]

[a] Electrical and Computer Engineering Dept., University of Rochester, Rochester, NY, 14627-0126, USA
[b] Xerox Corporation, 800 Phillips Road, Webster, NY, 14580, USA
[c] College of Engineering, Koc University, Istanbul, Turkey
celik@ece.rochester.edu, g.sharma@ieee.org, tekalp@ece.rochester.edu, eli.saber@usa.xerox.com

ABSTRACT

We present a novel reversible (lossless) data hiding (embedding) technique, which enables the exact recovery of the original host signal upon extraction of the embedded information. A generalization of the well-known LSB (least significant bit) 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 static portions of the host as side-information improves the compression efficiency, and thus the lossless data embedding capacity.

1. INTRODUCTION

Most multimedia data embedding techniques modify, and hence distort, the host signal in order to insert the additional information. Often, this *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 *Reversible (Lossless) Data Embedding* techniques. These techniques, like their lossy counterparts, insert information bits by modifying the host signal, thus induce 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.

Lossless data embedding techniques may be classified into one of the following two categories: Type I algorithms [1] 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 watermark signal is removed, i.e. subtracted, to restore the original host signal. Potential 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. Payload extraction in Type-I algorithms is robust. On the other hand, modulo arithmetic may cause disturbing salt-and-pepper artifacts.

In Type II algorithms [2, 3], information bits are embedded by modifying, e.g. overwriting, selected features (portions) of the host signal -for instance least significant bits or high frequency wavelet coefficients-. Since the embedding function is inherently 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 original host signal is restored by replacing the modified features with the decompressed original features. In general, Type II algorithms do not cause salt-and-pepper artifacts 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, we will introduce a generalization of the well-known LSB (least significant bit) modification method as the underlying irreversible (lossy) embedding technique. This technique, modifies the lowest levels- instead of bit planes- of the host signal to accommodate the payload information. In the second part, a lossless data embedding algorithm for continuous-tone images is built 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. This property of the proposed method provides excellent compression of relatively simple image features. Earlier algorithms in the literature tend to select more complex features to improve the compression performance- thus the lossless embedding capacity-.

2. GENERALIZED LSB EMBEDDING

One of the earliest data embedding methods is the LSB (least significant bit) modification. In this well-known method, the LSB of each signal sample is replaced (over-written) by a payload data bit. During extraction, these bits are read in the same scanning order, and payload data is reconstructed. A generalization of the LSB embedding method is employed here. If the host signal is represented by a vector $\mathbf{s}$, the generalized LSB embedding and extraction processes can be represented as

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

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

where $\mathbf{s}_w$represents the signal containing the embedded information, $\mathbf{w}$ represents the embedded payload vector of L-ary symbols,

i.e. $w_i \in \{0, 1, ..., L-1\}$, and $Q_L(x) = L\lfloor \frac{x}{L} \rfloor$ is an L-level scalar quantization function.
In the embedding phase, the lowest L-levels of the signal samples are replaced (over-written) by the watermark payload. During extraction, watermark payload is extracted by obtaining the quantization error- or simply reading lowest L-levels- of the watermarked signal. The classical LSB modification is a special case where $L = 2$. Generalized LSB embedding enables embedding of non-integral number of bits in each signal sample and thus introduces new operating points along the rate (capacity)-distortion curve.

2.1. 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, a binary to L-ary (and L-ary to binary) pre(post) conversion utility is required. Moreover, in practice signal values are generally represented by finite number of bits, which can afford only a limited range of sample values. In certain cases, 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 M values (w is an M-ary symbol) where $M \leq L$.
In order to address these concerns, we employ the following algorithm which converts binary input $\mathbf{h}$ into a L-ary symbols while preventing over-flows. We start by interpreting the binary input string as the binary representation of a number H in the interval $[0, 1)$, i.e. $H = .h_0h_1h_2...$ and $H \in [0, 1)$. Furthermore, we let R represent this interval ($[0, 1)$).

1. Given s and $Max(s)$, determine $Q_L(s)$ and number of possible levels $M \leq L$,
2. Divide R into M equal sub-intervals, R_0 to R_{M-1}
3. Select the sub-interval that satisfies $H \in R_m$
4. Next watermark symbol is $w = m$
5. Set $R = R_m$ and goto step 1, for the next sample

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), M, on a given signal sample s_w are obtained from $Q_L(s_w) = Q_L(s)$.

2.2. Embedding Capacity and Distortion

In Generalized-LSB embedding (Eqn. 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 $C_{GLSB} = log_2(L)$ bits per sample (bps).
A closed form expression for the expected mean square and mean absolute error distortions may be obtained if we assume that: *i)* data symbols $\mathbf{w}$ are equiprobable, which is reasonable if input data is compressed and/or encrypted, as in many data embedding applications; and *ii)* 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.

$$D(MSE) = \frac{1}{L^2}\sum_{r=0}^{L-1}\sum_{w=0}^{L-1}(r-w)^2 = \frac{L^2-1}{6} \quad (3)$$

$$D(MAE) = \frac{1}{L^2}\sum_{r=0}^{L-1}\sum_{w=0}^{L-1}|r-w| = \frac{L^2-1}{3L} \quad (4)$$

3. LOSSLESS GENERALIZED-LSB DATA EMBEDDING

The G-LSB embedding algorithm 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.
Fig. 1 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 (Eqn. 5). 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, resulting bit-stream is converted to L-ary symbols $\mathbf{w}$ and added to the quantized host to form the watermarked signal $\mathbf{s}_w$ (Eqn. 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 (Eqn. 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 (Eqn. 6).

$$\begin{aligned} \mathbf{r} &= \mathbf{s} - Q_L(\mathbf{s}) \quad (5) \\ \mathbf{s} &= Q_L(\mathbf{s}) + \mathbf{r} = Q_L(\mathbf{s}_w) + \mathbf{r} \end{aligned}$$

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 capacity. The lossless embedding capacity of the system is given by,
$C_{Lossless} = C_{GLSB} - C_{Residual}$. This observation emphasizes the importance of the residual compression algorithm.

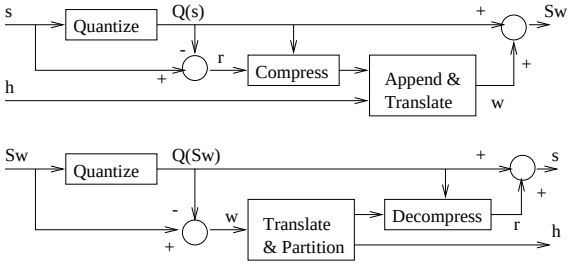

Fig. 1. Embedding (top) and extraction (bottom) phases of the proposed lossless data embedding algorithm.

3.1. 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 (Eqn. 5), 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 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 [4] for the lossless embedding scenario. The algorithm comprises of three main components: *i)* prediction, *ii)* context modeling and quantization, *iii)* conditional entropy coding. The prediction step 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 pseudo-code:

1. $\hat{s}_O$ = Predict Current Pixel();
2. (d, t) = Determine Context D,T($\hat{s}_O$);
3. $\dot{s}_O$ = Refine Prediction($\hat{s}_O, d, t$);
4. m = Determine Context M($\dot{s}_O$);
5. If ($m \geq 0$), Encode/Decode Residual(r_O, d, m);
   else, Encode/Decode Residual($L - 1 - r_O, d, |m|$);

3.1.1. Prediction

Let us assume that the residual samples, r, are encoded and decoded in the raster scan order and denote a pixel position and its 8-connected neighbors by their relative directions, i.e. O, W, NW, N, NE, E, SE, S, SW, respectively. The prediction uses the quantized pixel values, $Q_L(s)$, at these positions and additionally the already reconstructed residual in the causal neighborhood (W,NW,N,NE). 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 $\frac{L}{2}$ to compensate for the bias in the truncation $Q_L(.)$).

$$f(s_k) = \begin{cases} s_k & \text{if } k \in \{W, NW, N, NE\} \\ Q_L(s_k) + \frac{L}{2} & \text{o/w} \end{cases} \quad (6)$$

An initial linear prediction of the current pixel value based on 4-connected neighbors is given by,

$$\hat{s}_O = \frac{1}{4} \sum_{k \in \{W,N,E,S\}} f(s_k) \quad (7)$$

However, this predictor is often biased, resulting in a non-zero mean for the prediction error, $s - \hat{s}$. As in [4], we refine this prediction and remove its bias using a feed-back loop, on a per-context basis. The new prediction is calculated as,

$$\dot{s}_O = \lfloor \hat{s}_O + \bar{\epsilon}(d, t) \rfloor \quad (8)$$

where $\bar{\epsilon}(d, t)$ is the average prediction error ($\epsilon = s_O - \dot{s}_O$) of all previous pixels in the given context (d, t).

3.1.2. Context Modeling and Quantization

Typical natural images exhibit non-stationary 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, may be exploited in encoding the corresponding pixels. If the contexts and the corresponding statistical models are chosen appropriately, this process can yield significant improvements in coding efficiency.

We adopt a variant of d and t contexts from [4]. These contexts correspond to local activity and texture measures.

$$\Delta = \sum_{k \in \{W,NW,N,NE,E,SE,S,SW\}} \frac{1}{8} |f(s_k) - \hat{s}_O| \quad (9)$$

$$d = Q(\Delta) \quad (10)$$

$$t_k = \begin{cases} 1 & \text{if } f(s_k) > \hat{s}_O \\ 0 & \text{o/w} \end{cases} \quad (11)$$

$$t = t_W \| t_N \| t_E \| t_S \quad (12)$$

where t is obtained by concatenating t_k bits (16 values), and $Q(\Delta)$ is a scalar non-uniform quantizer with 8 levels. The thresholds are determined experimentally as $\{1, 2, 3, 4, 6, 10, 15\}$, to include approximately equal number of pixels in each bin.

Once these (d, t) contexts are determined, prediction is refined as in Eqn. 8. Typically the prediction error, $\epsilon = s - \dot{s}$, will have Laplacian statistics with zero mean and a small variance. Given $\dot{s}$, distribution of pixel values $p(s = \dot{s} + \epsilon|d)$ is similar to the prediction error distribution $p(\epsilon|d)$. It will have a peak at $\dot{s}$, and decreases with increasing distance from $\dot{s}$. Moreover, given $Q_L(s)$, limits s to the range $[Q_L(s), Q_L(s) + L)$, and when normalized $p(s|d)$ in this range gives the probability distribution of the corresponding residuals, $r \in [0, L)$.

A third context, m, groups each residual according to the shape of the its probability distribution. This shape is mainly determined by the position of its peak (see Fig. 2). If $Q_L(s) > \dot{s}$, the peak is at $r = 0$ and the distribution monotonically decreases. Likewise, if $Q_L(s) + L - 1 < \dot{s}$, the peak is at $r = L - 1$ and the distribution monotonically increases. Since the first is a mirror image of the latter, it can be eliminated by re-mapping (flipping $r_{new} = L-1-r_{old}$) its values prior to coding. This information is encoded in the sign of m, where magnitude of m is kept constant. In other cases, when $Q_L(s) \leq \dot{s} \leq Q_L(s) + L - 1$, the peak is at $r = \dot{s} - Q_L(\dot{s})$. Due to the symmetry of the Laplacian, distributions having peaks at r_p and $L - 1 - r_p$ are mirror images and same reduction may be applied (assign $\pm m_i$).

3.1.3. Conditional Entropy Coding

At 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. For each coding context (d, m), conditional probabilities of residuals are estimated from the previously encoded(decoded) samples.

4. EXPERIMENTAL RESULTS

The proposed algorithm was tested on the uncompressed 512×512 gray-scale images seen in Fig. 3. Table. 1 and Fig. 4 show

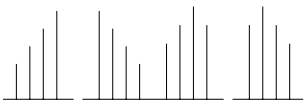

Fig. 2. PMFs of r for contexts $m = \{\pm 1, \pm 2\}$ ($L = 4$).

the available lossless data embedding capacity (in Bytes (x8 bits)) obtained for various embedding strengths (levels L).

Fig. 3. Test set: 512x512 gray-scale images

Level(L)	2	3	4	5	6
PSNR(db)	51.1	46.9	44.2	42.1	40.5
F-16	2223	4823	7685	10205	13479
Mandrill	83	248	459	753	1111
Boat	632	1703	3055	4578	6161
Barbara	561	1507	2689	4073	5525
Gold	310	882	1575	2448	3434
Lena	601	1543	2848	4286	5890
Level(L)	8	10	12	14	16
PSNR(db)	38.0	36.0	34.4	33.0	31.9
F-16	17877	22675	26860	30742	34083
Mandrill	1897	2796	3821	4603	5751
Boat	9783	13122	16272	18611	22225
Barbara	8264	11140	13624	16158	17593
Gold	5627	7955	10403	12328	14553
Lena	9325	12680	15774	19137	22130

Table 1. Lossless Embedding Capacity (in Bytes) vs. embedding levels(L) and average PSNR(dB) at full capacity

In Fig. 4, we see that the capacity of the proposed method depends largely 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. These distributions result in shorter code-lengths, and thus higher embedding capacities.

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. The rate of the increase, however, is not constant either among images or throughout the levels.

Note that the embedding capacities illustrated in Fig. 4 are achieved because the conditional entropy coding scheme adopted here successfully exploits the intra pixel correlation among the different

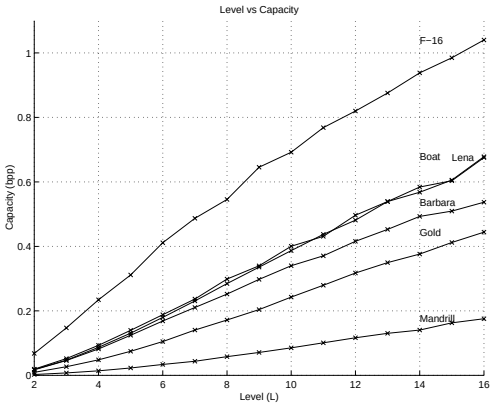

Fig. 4. Capacity $C_{Lossless}$ vs Levels for all images

levels of the same pixel and the inter-pixel correlations among neighbors. A direct compression approach that attempts to compress the residual signal alone without utilizing the rest of the image performs significantly worse. For instance, the context-less approach requires an embedding level $L \geq 15$ in order to achieve capacities comparable to the presented scheme at $L = 2$. The higher embedding level implies significantly higher distortion in the watermark bearing signal.

5. 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 level to meet the capacity requirements, thus trading off intermediate distortion for increased capacity. In such scenarios, the generalized LSB embedding proposed in the current paper is significantly advantaged over conventional LSB embedding techniques because it offers finer granularity along the capacity distortion curve.

6. REFERENCES

[1] C.W. Honsinger, P.W. Jones, M. Rabbani, and J.C. Stoffel, "Lossless recovery of an original image containing embedded data," *US Pat. #6,278,791*, Aug 2001.

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

[3] J. Tian, "Wavelet-based reversible watermarking for authentication," *Proc. of SPIE Sec. and Watermarking of Multimedia Cont. IV*, vol. 4675, no. 74, Jan 2002.

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