IMPROVED TECHNIQUES FOR WATERMARKING HALFTONE IMAGES

Phil Sherry and Andreas Savakis

Rochester Institute of Technology
Department of Computer Engineering
pes5750@rit.edu, savakis@mail.isc.rit.edu

ABSTRACT

Watermarking schemes that have been developed for continuous tone images cannot be directly applied to halftone images. Many of these watermarking methods require characteristics implicit in continuous tone images but absent from halftones. With this in mind, it seems reasonable to have a robust watermarking technique specific to halftones, equipped to work in the binary image domain. This paper reviews existing techniques and suggests improvements to increase performance and overcome limitations of existing techniques. The first set of post-halftone watermarking methods work on existing halftones. Data Hiding Cell Parity (DHCP) works in the parity domain instead of individual pixels. Data Hiding Mask Toggling (DHMT) works by encoding two bits in the 2x2 neighborhood of a pseudorandom location. Dispersed Pseudorandom Generator (DPRG), on the other hand, is a preprocessing method that takes place before halftoning. DPRG uses a dispersed pseudorandom generator that achieved better visual results. Using the Modified Peak Signal-to-Noise Ratio (MPSNR) metric, the proposed techniques outperform existing methods by up to 5-15%, depending on the image type and method considered.

1. INTRODUCTION

Watermarking is a common image processing operation, used for authentication, tracking, copyright control, and many other purposes. Watermarks can be broken down into two types: robust and fragile. Fragile watermarks are used primarily for authentication and are easily exposed if tampered with. Robust watermarks are intended to be invisible to the naked eye under all circumstances. Such watermarks are designed to be invariant to rotation, stretching, cropping, translation, noise, and all other degradations.

While many techniques exist for embedding data in continuous tone images, they are not always reliable when used directly with halftone images [10]. Halftone images, those which are represented with binary data, require their own class of watermarking techniques. This paper analyzes some existing halftone techniques and proposes enhancements to improve their performance.

2. EXISTING TECHNIQUES

The halftone watermarking methods described within are proposed improvements to existing techniques [1], designed to improve visual quality and/or computational efficiency. This is of great value for other watermarking techniques [7] based upon [1]. Since the methods within are simple and operate on the binary data directly, they have a significant advantage over other halftone watermarking schemes, which require a separate key image for watermark extraction [4;5] or are computationally intensive [6].

For the sake of image quality, it is desirable to have the original continuous tone image present when creating the watermarked image. However, there may exist situations where only the halftone is available. The methods described in this paper are tailored to suit the specific needs of each case.

2.1. Data Hiding Self Toggling (DHST)

The simplest watermarking method is data hiding self toggling (DHST), which works on the halftone image [1]. Encoding works by choosing pseudorandom candidate pixels and setting to the bit values of the data stream. Unfortunately, toggling individual pixels creates disturbances in the local grey level, causing noticeable salt-and-pepper artifacts. By using the same pseudorandom seed for encoding and decoding, the embedded data can be easily retrieved. For decoding, the only information that must be known is the seed, image dimensions (when the halftone is scanned), and the data length. Most of the subsequent watermarking schemes attempt to mitigate the visual problems introduced by DHST while retaining its decoding simplicity. Unless otherwise noted, DHST is used for decoding of all methods.

2.2. Data Hiding Smart Pair Toggling (DHSPT)

Data hiding pair toggling (DHSPT) seeks to maintain the local greylevel by performing two complementary toggles, when necessary [1]. If the DHST process causes a pixel to toggle, a complementary "slave" pixel is toggled from the 3x3 neighborhood (slave candidates are opposite to the current pixel). DHSPT uses a deterministic method to find the best complementary toggle. For all slave candidates, (1) is used to calculate a pixel's connectedness. The more pixels in the slave's 3x3 neighborhood that have the same color as the reference x_0 (i.e. the slave pixel), the higher x_0's connectedness.

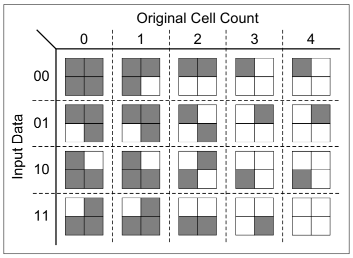


Fig. 1. DHMT Encoding Scheme.

T	T	T	T
T	P	P	T
T	P	P	T
T	T	T	T

Fig 2. DHCP Embedding Locations.

$$DHCP\ Parity = \left(\sum P\right) \bmod 2 \tag{2}$$

The weighting factor w(i)=[1 2 1; 2 0 2; 1 2 1] in the neighborhood [x_1 x_2 x_3; x_4 x_0 x_5; x_6 x_7 x_8] gives adjacent pixels a greater weight since they tend to look more “connected”. When assessing the slave candidate pixels, it is desirable to find the one with the lowest degree of connectedness, as connected pixels are much more noticeable.

$$con(m,n) = \sum_{i=1}^{8} w(i) f(x_0, x_i),\quad f(x,y) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases} \tag{1}$$

2.3. Modified Data Hiding Error Diffusion (MDHED)

The second set of watermarking techniques assumes that the original greylevel image is available. This provides some flexibility in terms of encoding because toggling errors can be dispersed amongst multiple pixels. Modified data hiding error diffusion (MDHED) [1] works by first performing DHST followed by regular error diffusion [8]. The diffusion step compensates for the hard coded data pixels. Since the DHST stage fixes the pixels before diffusion, the data pixels remain the same in the halftone. Also, the DHST error is diffused to neighboring pixels in both the forward and backward directions. This is possible because DHST distributes the embedding candidates randomly across the image, meaning that there are contone pixels on all sides available to receive error. The amount of error that is diffused to the feedback and feedforward pixels is determined by a scalar α that ranges from 0 to 1. Note that the forward and backward kernels can be of different size and weight (i.e. Jarvis for feedback and Steinberg for feedforward[9]).

3. PROPOSED HALFTONE WATERMARKING IMPROVEMENTS

The following watermarking techniques have been devised to improve upon the performance of those described above, whether that means improved image quality, greater efficiency, or increased data hiding capability.

3.1. Data Hiding Cell Parity (DHCP)

Data hiding cell parity (DHCP) works on existing halftones by encoding the data stream in the parity domain instead of individual pixels. The DHCP parity is contained within the 2x2 neighborhood to the bottom right of the current data pixel defined by the DHST pseudorandom generator (Fig. 2). For each bit in the data stream, if that bit is equal to the DHCP parity (2), no action is performed. Otherwise, a complementary toggle is performed. This is done in similar fashion to DHSPT, except that the slave pixel must be one of the bordering T pixels (Fig. 2). Note that the actual value of the P pixels is unimportant. As long as the T slave pixel is opposite to the P master pixel, it is a viable toggling candidate. As in DHSPT, the pixels chosen for toggling are the ones with the lowest post-toggling connectedness. DHCP decoding is performed by using the DHCP parity function (2) in the 2x2 DHCP neighborhood.

The main motivation for DHCP is the increased DHSPT slave candidate search space. DHCP raises the maximum number of slave candidates from 8 to 20. While the image quality should definitely improve with DHCP, the encoding and decoding speeds will suffer. During encoding, there are 2.5 more connectedness functions to compute (20 vs. 8). Also, for both encoding and decoding, the parity must be computed for the 2x2 DHCP region. This can be accomplished with just a few XOR gates, but must be taken into account nonetheless.

3.2. Data Hiding Mask Toggling (DHMT)

Data hiding mask toggling (DHMT) is a post-halftoned watermarking method that works by encoding two bits in the 2x2 neighborhood of the pseudorandom location generated by DHST (upper left pixel is DHST data pixel, the same region as DHCP). Each 2-bit data combination is assigned a set of five possible encoding masks (Fig. 1). The mask used for encoding depends on the cell count of the original image cell. The cell count is defined as the number of white pixels in the original 2x2 cell, with an all black cell having a count of zero, an all white cell having a count of four, and other combinations in between. The encoding masks were chosen in order to maintain the cell count, when possible, maintaining local 2x2 cell intensity and minimizing visual degradation. In the event that the cell count cannot be maintained, the mask with the closest cell count is used. Inspecting Fig. 1, the cell count cannot be maintained when the original cell is all black and the input data is “11”. In this case, the mask for the cell count of 1 is used (upper left white pixel). Because each mask is specific to a certain 2-bit combination, decoding is fairly simple. Retracing the DHST pseudorandom locations used for encoding, each 2x2 cell is compared to the table in Fig. 1. The 2-bit combination that matches the cell is the data. Thus, watermark decoding is accomplished with a simple 16-entry look-up table, making DHMT very computationally efficient. Compare this to DHSPT, which uses 8x2 con operations.

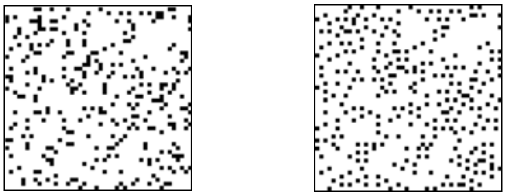


Fig. 3. DHST and DPRG Embedding Locations. DHST (left) clusters the data locations much more than DPRG (right).

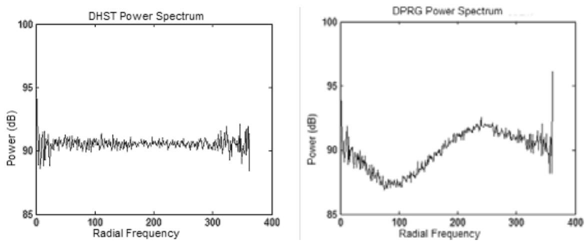


Fig. 4. DHST and DPRG Radially Averaged Power Spectra.

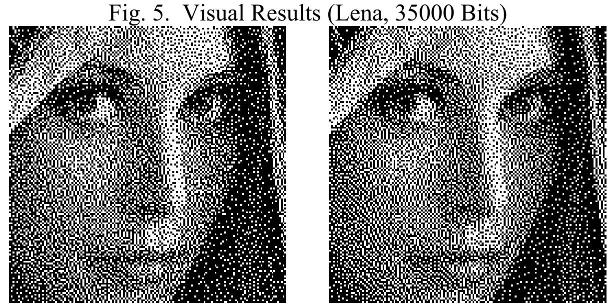
3.3. Dispersed Pseudorandom Generator (DPRG)

The dispersed pseudorandom generator (DPRG) differs from the preceding watermarking improvements in that it is a preprocessing step. Data embedding locations are normally chosen using a uniform pseudorandom generator. However, this leads to clustering of embedding locations, especially at high data rates. MDHED relies on the fact that the error diffusion stage can compensate for the embedding distortion. Unfortunately, embedded data pixels cannot receive diffused error, and by clustering these pixels, a larger amount of error is unaccounted for. DPRG works by reducing the connectedness of the data embedding locations (Fig. 3). When a data location is chosen, its connectedness to other embedding locations is calculated using the con function defined by DHSPT. If the connectedness of that location is not zero, the pixel in the 3x3 neighborhood with the lowest connectedness is used instead. Fig. 4 shows how the power spectrum of the DPRG embedding displays a better blue noise characteristic than regular DHST embedding [9]. Although not ideal, the DPRG spectrum does concentrate more power in higher frequencies. Decoding is performed by following the same steps as encoding. The encoding and decoding complexities are about the same as if DHSPT were used in lieu of DHST. That is, up to 8 extra connectedness functions could be performed for each data location.

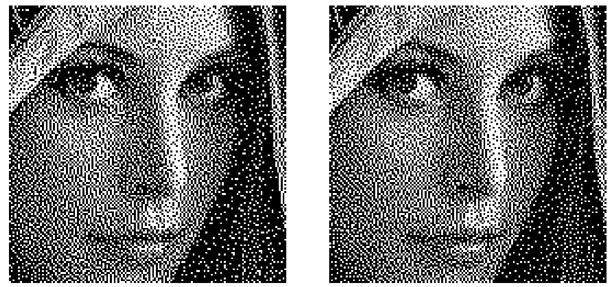
4. RESULTS AND DISCUSSION

Watermarking performance was measured by embedding data in standard test images in rates ranging from 100 to 100000 bits. The image quality was tracked using a metric known as MPSNR (described below). When compared against a baseline (the MPSNR of the original halftone), the scalability of each method was determined. From a practical standpoint, a user might want to maintain a certain post-watermarked image quality level. By choosing a minimum allowable MPSNR, the maximum embedding rate for any given method can be derived from the graphs.

Fig. 5. Visual Results (Lena, 35000 Bits)



(a) DHSPT (MPSNR=24.15) (b) DHCP (MPSNR=24.42)



(c) MDHED (MPSNR=25.24) (d) DPRG (MPSNR=25.46)

4.1. Modified Peak Signal to Noise Ratio (MPSNR)

In order to facilitate the development of improved data hiding methods, a halftone image quality metric was needed. Typically, halftone quality is measured by just "eyeballing" the orignal and halftone. The de-facto contone image quality metric is peak signal to noise ratio (4), derived from root mean squared error (3) [2]. F and H represent the contone and halftone, respectively, with dimensions of NxM. A higher PSNR denotes better image quality. However, PSNR fails when comparing the original greyscale and halftone. This is because individual pixels in the halftone do not correlate to pixels in the greyscale. The human visual system acts like a low-pass filter, blurring neighboring pixels to approximate a continuous-tone image. In order to obtain a reliable quality metric this phenomenon must be simulated.

Modified peak signal to noise ratio (MPSNR) is a quality metric that attempts to model the human visual system. First, a simple inverse halftone H_{low} is generated with a low pass filter. For our purposes, a 5x5 Gaussian low-pass filter worked nicely [3]. H_{low} is then fed into the regular PSNR function to generate the MPSNR (5). This function allows for automated algorithm testing. Note that MPSNR, like PSNR, measures relative visual quality, meaning that MPSNR measurements can only be compared between the variations of same image.

$$RMSE(F,H) = \sqrt{\frac{\sum\left[F(i,j) - H(i,j)\right]^2}{N \cdot M}} \tag{3}$$

$$PSNR(F,H) = 20\log\frac{255}{RMSE(F,H)} \tag{4}$$

$$MPSNR(F,H) = PSNR(F,H_{low}) \tag{5}$$

TABLE I
MPSNR OF HALFTONE WATERMARKING TECHNIQUES

Method	Bits Embedded	Lena	Boat	Frog	Peppers	Avg.
Baseline		26.32	26.24	23.96	26.72	25.81
DHSPT	5000	25.89	25.75	23.69	26.27	25.40
DHMT	5000	25.92	25.84	23.75	26.32	25.46
DHCP	5000	25.91	25.78	23.71	26.29	25.42
MDHED	5000	26.22	26.11	23.91	26.57	25.70
DPRG	5000	26.23	26.11	23.91	26.59	25.71
DHSPT	20000	24.88	24.67	23.05	25.22	24.46
DHMT	20000	24.84	24.79	23.20	26.21	24.51
DHCP	20000	25.03	24.75	23.09	25.38	24.56
MDHED	20000	25.77	25.65	23.75	26.15	25.33
DPRG	20000	25.88	25.71	23.77	26.21	25.39
DHSPT	35000	24.15	23.95	22.58	24.53	23.80
DHMT	35000	23.96	23.99	22.68	24.31	23.74
DHCP	35000	24.42	24.04	22.62	24.77	23.96
MDHED	35000	25.24	25.17	23.55	25.68	24.91
DPRG	35000	25.46	25.31	23.59	25.80	25.04

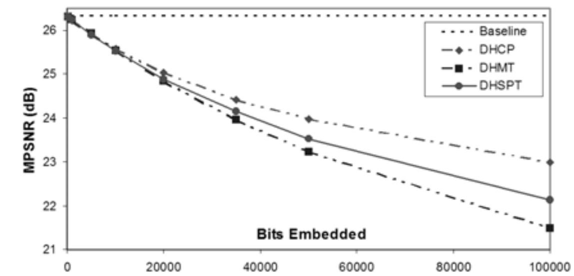


Fig. 6. Post-Halftone Watermarking Performance (Lena)

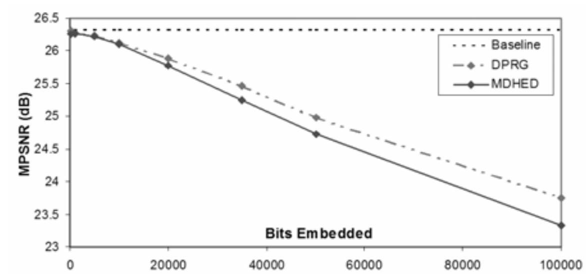


Fig. 7. Pre-Halftone Watermarking Performance (Lena)

4.2. Watermarking Performance

The post-halftone watermarking performance is shown in Fig. 6. DHMT exhibits the lowest performance of the proposed methods. The image quality of DHMT is highly dependent on the sample image and embedding rate. The biggest advantage of DHMT, as described above, is that the encoding complexity is lower than the other methods. The performance is actually quite high when fewer bits are embedded. The degradation can be attributed to the fact that the pixel toggling isn't as selective as DHSPT, which uses the connectedness function. DHCP is the clear winner in terms of image quality at high embedding rates. Its complexity is justifiable since it produces the best post-halftoned results.

Fig. 7 shows the performance of DPRG versus MDHED. The results for DPRG are quite good, lending credence to the blue noise embedding pattern. With error diffusion, the more the error can be diffused, the better. The added complexity of DPRG can be justified by the fact that no other watermarking method can produce such high quality images. DPRG MPSNR values are higher than MDHED by about 0.1-0.2, which translates to embedding rate increases of 5-15%.

Table I shows a comparison of the various methods with different test images. One item of note is DHMT's widely varying performance. As mentioned above, DHMT image quality is quite good at low embedding rates due to the fact that it embeds two bits per step. Only at high rates does the lack of a toggling heuristic damage the performance. Fig. 5 shows visual samples of select watermarking methods using an embedding rate of 35000 bits. Notice how the best performing method, DPRG, fixes some of the anomalies present in MDHED, especially around the nose and hat brim.

5.

CONCLUSION

As indicated by the experimental results, DHCP and DPRG are viable improvements to existing halftone watermarking methods (DHSPT and MDHED). These techniques provide increased data hiding capabilities for creating post- or pre-halftone watermarks, respectively. DHMT was found to be a possible replacement for DHSPT, but only with certain images and at low data embedding rates.

REFERENCES

[1] Ming Sun Fu and Oscar C. Au, "Data Hiding Watermarking for Halftone Images," in *IEEE Transactions on Image Processing*, Vol. 11, No. 4, April 2002, pp. 477-484.
[2] Larry Rowe, "Image Quality Computation," University of California, http://bmrc.berkeley.edu/courseware/cs294/fall97/assignment/psnr.html
[3] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing, 2nd Edition," Prentice-Hall, Inc. 2002.
[4] "Digital Watermarking in Printed Images: Stochastic Halftone Screen Design," in XRCE Texnology Showroom Fact Sheet, Xerox Corportation.
[5] Ming Sun Fu and Oscar C. Au, "Self-Conjugate Watermarking Technique for Halftone Images," Electronics Letters, Vol. 39, Issue 4, Feb 2003.
[6] Zachi Baharav and Doron Shaked, "Watermarking of Dither Halftoned Images," HP Laboratories Israel, Hewlett-Packard Company, May 1999.
[7] Hae Yong Kim and Amir Afif, "Secure Authentication Watermarking for Binary Images," Universidade de Sao Paulo, Sao Paulo, Brasil.
[8] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," in Proc. SID, 1976, pp. 75-77.
[9] Robert Ulichney, *Digital Halftoning*, The MIT Press, Cambridge, Massechusetts, pp 2-3, 54-61, 77-186, 1987.
[10] Murat Mese and P.P. Vaidyanathan, "Effects of Inverse Halftoning in Watermarking," Department of Electrical Engineering, California Institute of Technology.