Embedding Color Watermarks into Halftoning Images using Minimum-Distance Binary Patterns

Pedro Garcia Freitas*, Mylène C.Q. Farias† and Aletéia P. F. de Araújo‡
*‡Department of Computer Science,
†Department of Electrical Engineering,
University of Brasília, Brazil
Email: *sawp@sawp.com.br, †mylene@ieee.org, ‡aleteia@cic.unb.br

***Abstract*—This paper presents a halftoning-based watermarking method. This method enables the embedding of a color image into a binary black-and-white halftone, while maintaining the image quality. The proposed technique is capable of embedding watermarks of three color channels into a binary halftone. To achieve high quality halftones, the method maps colors to halftone channels with homogeneous dot patterns which in turn use different binary texture orientations to carry the watermark. They are obtained by solving a minimization problem in which the objective function is the binary distance between the original binary halftone and the available patterns. To restore the color information, we scan the printed halftone image and compute the inverse information (considering the dot pattern). Using the mapped information, we restore the original color channels from the halftone images using a high-quality inverse halftoning algorithm. Experimental results show that the method produces restorations with a superior quality than other methods found in the literature and increases the embedding capacity.**

***Keywords*-Embedding, Halftone, Color Restoration, Watermark, Inverse Halftoning.**

I. Introduction

Transmitting side information using printed media is a challenge due to the distortions introduced by the print-and-scan (PS) process [1]. Some of these distortions occur because the displayed digital color may differ from its printed representation. One of the causes for these differences is the fact that the digital image is converted to a halftone representation before being printed [2]. This halftone representation is generated from a mathematical model that produces colors using a combination of colored dot patterns [3]. The halftone images are perceived as continuous tone images when viewed from a distance due to a low-pass property of the Human Visual System (HVS). Many different halftoning methods have been developed, including Direct Binary Search (DBS) [4, 5], Ordered Dithering (OD) [6, 7], Error Diffusion (ED) [8]–[10], and Dot Diffusion (DD) [11]–[13]. Although there is a great diversity of image halftoning methods, most of them modify the coding information of the printing process.

The scanning process performs the inverse task of the printing process. Scanner devices read the printed halftone and restore a multi-level image via an inverse halftoning algorithm [14, 15]. Although the inverse halftoning algorithm recovers the distinct intensity levels of the original image, the reconstructed image may present distortions like noise [16] and blur [17]. Therefore, PS processes make hardcopy watermarking more challenging than digital watermarking.

Many works in the literature address the document hardcopy problem by trying to keep the hidden information on a PS channel more robust. Most prior work on image data hiding target color and grayscale images with a wide range of intensity values [18]–[20]. These methods differ from each other in terms of efficiency, capacity, and robustness. Brassil *et al.* [21] propose authentication methods based on shift coding. To increase the robustness, their methods require the use of uniformly spaced centroids, which are often difficult to obtain. Tan *et al.* [22] extended these methods using a directional modulation technique for watermarking of Chinese text images. More recently, other methods were proposed for specific applications [23]–[26].

Among the available methods, those that embed information into binary images are very promising because the pixel binarization is the last step process of the printing process. When the scanner reads the paper, the data is first collected as a binarized halftone which increases the robustness of PS process. It is worth pointing out that, since binary images have less capacity to hide information, embedding data in binary images is more difficult than in color or grayscale images [27]. Although more difficult, the demand for this kind of technique is increasing and several binary-image watermarking techniques have been developed [28]–[32]. However, as stated by Hou *et al.* [27], these methods have several limitations that include a limited data capacity and the presence of noticeable artifacts.

Some approaches have been proposed to increase the embedding capacity of binary-images embedding. Pan *et al.* propose a low-capacity watermarking scheme for halftone image authentication, exploiting an image hash as a fragile watermark [33]. Guo and Liu [34] developed a higher capacity watermarking technique that uses a block truncation code. Son and Choo [35] proposed a watermarking method for clustered halftone dots in which the embedded binary data is recovered using dictionary learning. Guo *et al.* [36] proposed a halftoning-based approach capable of embedding watermarks using direct binary search to encode the binary data. Guo and Liu [37] propose a method for embedding a multi-tone watermark that, as a consequence, produces a lower quality image. Although all these methods have a reasonable embedding data capacity, they are restricted to a specific type of dithering that limits their performance and application.

More recently, Son *et al.* proposed some techniques [38, 39] to restore color channels from a black-and-white halftone image that has homogeneously distributed dot patterns. This work represents an improvement for the reversible color-to-grayscale conversion problem [40]–[42]. This conversion problem is specific to watermarking techniques that aim to recover the original color channels from watermarked grayscale images submitted to a PS process. This application involves a large amount of embedded data because it inserts two chrominance images (color channels) into a luminance channel (grayscale image) [43].

Given these halftoning and watermarking challenges, we propose a binary image watermarking method that uses minimum distance of binary patterns. The method encodes color images into dithering patterns of halftone images with a similar goal to the work done by Son *et al.* [38, 39]. However, contrary to their algorithms, that only allow self-embedding of a color channels into its halftone version, the proposed algorithm is far more flexible and it is able to embed any content (another image or itself).

The rest of this paper is organized as follows. Section II gives a brief overview of binary vector dissimilarity measures. Section III describes the proposed method, detailing encoding (embedding) and decoding (recovering) of watermarks. The experimental results are presented in Section IV. Finally, the conclusions are drawn in Section V

II. Distance of Binary Patterns

A binary vector Z with n dimensions is defined as

$$Z = \{z_1, z_2, \cdots, z_n\}, \tag{1}$$

where $z_k \in \{0, 1\}$, $\forall\ k \in \{1, 2, \ldots, n\}$. Given two vectors $X \in \Omega$ and $Y \in \Omega$, where Ω is the set of all n-dimensional binary vectors, let S_{ij}, $\forall (i, j) \in \{0, 1\}$, be the number of matching occurrences of i in X and j in Y, at corresponding positions. As indicated by Zhang and Srihari [44], there exist several measures that can be used to evaluate the similarity, $S(X, Y)$, between X and Y. Among these similarity measures, the most common are Dice, Jaccard-Needham, Sokal-Sneath, Kulzinsky (Matching), Rogers-Tanimoto, Sokal-Michener, Russell-Rao, and Yule.

For each similarity measure there is an associated dissimilarity measure $D(X, Y)$. While some similarity measures are not normalized, i.e. their values are not necessarily in the interval $[0, 1]$, the dissimilarity measures are typically normalized [45]. In this paper, normalized dissimilarities are called distances when they are computed for a collection of raw observation vectors stored in a rectangular array.

III. Proposed Method

The proposed method consists of decomposing the RGB color channels of an image watermark (W) into three binary channels. Each color channel of W is treated as a grayscale image and a halftoning algorithm is used to generate $R = \{r_{ij}\}$, $G = \{g_{ij}\}$, and $B = \{b_{ij}\}$, where $r_{ij}, g_{ij}, b_{ij} \in \{0, 1\}$. Using a combination of each pixel of these binary channels, we map a binary mask that is used to encode the host halftone. The method, therefore, involves three steps: generation of masks, color encoding of the watermark, and restoration of the original color watermark (decoding). We describe these three steps.

A. Mask Generation

The encoding masks are generated by computing the finite n-ary Cartesian Product of the set X that is defined as:

$$X^n = \prod_{k=1}^{n} X = \{(x_1, \ldots, x_n) : \forall k \in \mathbb{N}_n^* : x_k \in X\}, \tag{2}$$

where X^n is a set of all ordered n-tuples $\{x_k\}$ and each element x_k is a basis element of X^n. For $X = \{0, 1\}$ we have a total of 2^n distinct n-tuples. Each tuple is equivalent to a binary vector (defined in Eq. 1) and is used to map the combinations of each pixel of a halftoned RGB channel. Since each pixel of the R, G, and B halftones has only two possible values, the result of this Cartesian product is a set of $2^3 = 8$ triplets.

Next, using Eq. 2, we compute a larger set of tuples to represent the distribution of points in the halftone. The higher the number of tuples, the larger is the distribution of distinct dots, which means that there are more options to represent the distribution of original pixels in the grayscale halftone (i.e. higher fidelity to unmarked halftone). On the other hand, the more distinct tuples there are, the more space is required to represent them. For simplicity, in this paper we adopt $n = 9$, which gives a total of 512 distinct nonuples. Since we have 8 distinct triplets and 512 distinct nonuplets, each triplet is used to map a subset of these nonuplets. More specifically, each triplet t_k maps a set of 64 distinct nonuplets $L_k = \{l_1^k, l_2^k, \cdots, l_{64}^k\}$. For our application, the nonuplets are geometrically distributed in 3×3 matrices called 'masks'.

B. Watermark Embedding (Encoding)

Once generated the masks from nonuplets, we use the set of triplets to encode the color information into the masks. First, we compute the halftone of each color channel independently, generating R, G, and B. For each pixel in these planes, we extract a triplet,

$$\tau_{i,j} = \{r_{ij}, g_{ij}, b_{ij}\}, \tag{3}$$

where $r_{ij}, g_{ij}, b_{ij} \in \{0, 1\}$. These extracted triplets are compared with the triplets used as key to map a set of masks (see previous section). Then, we chose the set of nonuples L_k such that $\tau_{i,j} = t_k$.

After each pixel of W is mapped into a set of binary masks, we have to decide which masks to choose. Since the masks are adjusted to 3×3 blocks, the replacement of each pixel with a mask implies a 3 times magnification. Therefore, we magnify the host image H by a factor of 3 and compute its

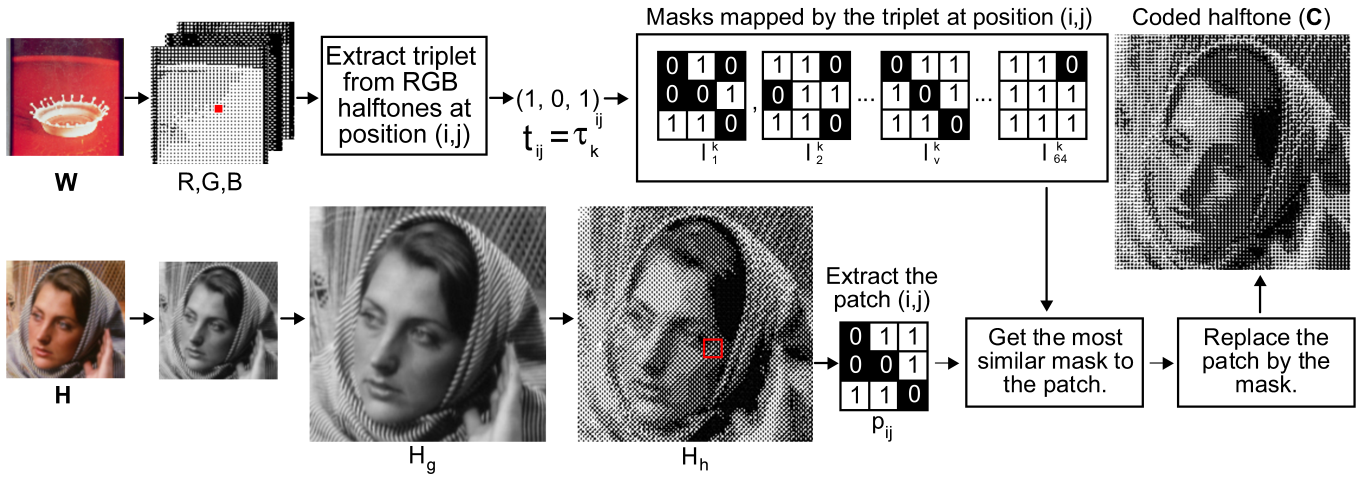

Fig. 1. Steps for embedding the color watermark into the halftone.

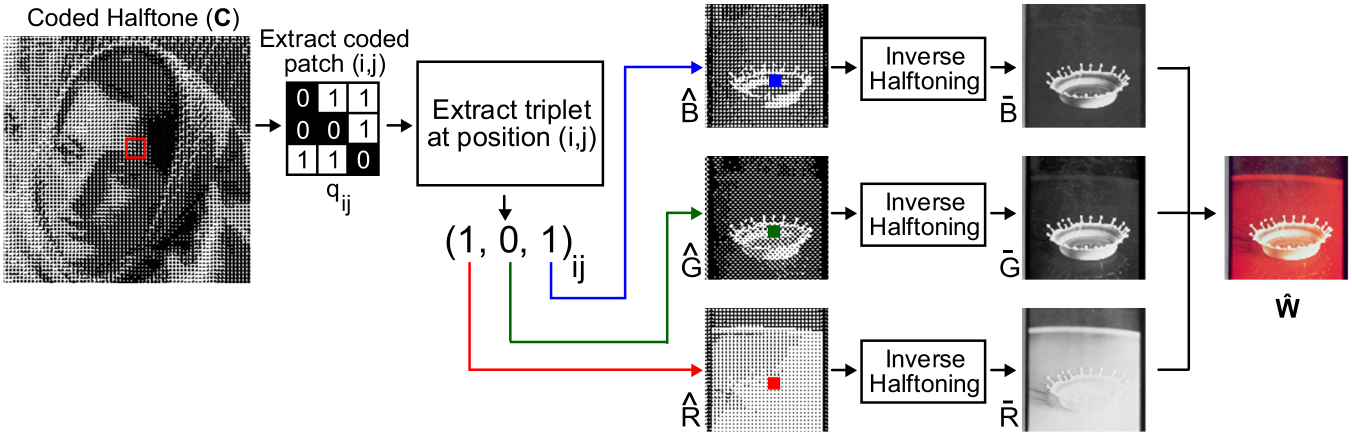

Fig. 2. Extraction of color watermark from encoded halftone.

halftone H_h from its grayscale H_g. This magnified halftone image is sliced into $M \times N$ patches p_{ij} of 3×3 pixels, where M and N are the dimensions of W.

After these steps, we have two matchable sets $P = \{p_{ij}\}$ and $\Xi = \{\tau_{i,j} \mapsto L_k\}$. For each p_{ij}, we choose the mask among the 64 available masks in each set by solving the following optimization problem:

$$\begin{aligned} \arg\min_{Y} \quad & D(X,Y) \\ \text{subject to} \quad & X = p_{ij} \\ & Y \in L_k, \end{aligned} \tag{4}$$

where D is the distance measured between X and Y. Since this minimization problem is performed for each patch, the encoded halftone image C is built by placing the computed masks at the corresponding position of the host halftone image (H_h). Fig. 1 depicts the steps of encoding the watermarked halftone C using the hidden information W and the host content H.

C. *Watermark Extraction (Decoding)*

The decoding process is depicted in Fig. 2. First, the encoded halftone C is sliced into $M \times N$ patches q_{ij} of 3×3 pixels. For each patch extracted from C, we recover the triplet $\hat{\tau}_{i,j}$ using the inverse mapping of L_k. In other words, we take $\hat{\tau}_{i,j} = t_k$ given that $t_k \mapsto L_k$, where $q_{ij} \in L_k$. In this manner, instead of applying the inverse halftone algorithm to recover grayscale levels, the information is extracted directly from the dot pattern of the printed image. Therefore, each restored triplet contains the bits of restored channels at position i, j:

$$\hat{\tau}_{i,j} = \{\hat{r}_{ij}, \hat{b}_{ij}, \hat{g}_{ij}\}, \tag{5}$$

where $\hat{r}_{ij}, \hat{g}_{ij}, \hat{b}_{ij} \in \{0, 1\}$. After these steps, we distribute the recovered bits to the respective halftone color channels $\hat{R}$, $\hat{G}$, and $\hat{B}$.

After recovering the halftones versions of the RGB channels, we use an inverse halftoning algorithm to restore the multilevel representation of these color channels: $\bar{R}$, $\bar{G}$, and $\bar{B}$. Finally, the color watermarked image $\hat{W}$ is restored after combining the restored channels.

IV. Experimental Results

The experiments were performed using a laptop with an Intel i7-4700MQ processor, 32GB of RAM, and a multi-functional Aficio MP C4501 (printer and scanner). The code is implemented in Python 3, with some extensions in Fortran 90 and Cython (to avoid performance issues). The proposed methods is tested using a set of 8 natural color images taken from the "Miscellaneous" set of USC-SIPI Image Database [46]. The original images are depicted in the first column of Fig. 3.

The proposed method is able to embed any content, related or not to the host image content. However, in our simulations we use the same image content for both image and host. The test consists of recovering the color channels from a printed halftone that was previously watermarked.

A. Performance of Color Restoration Method

The performance of the proposed method is compared with two state-of-the-art methods: Queiroz [40] and Ko *et al.* [41]. Since the proposed algorithm magnifies the original image by a factor of 3, in our experiments the grayscale images encoded with Queiroz's and Ko's methods are also magnified by a factor of 3. This way, all printed and scanned halftones have the same size for all tested methods.

We test the performance of the algorithms by Queiroz and Ko using three configurations for each method. In the first configuration, the images are not printed and scanned (NoPS). In the second configuration, the PS process is used. In the last configuration, the PS process is used in combination with a color enhancement (CE) algorithm to the restored color images. The scanned halftoning images are restored to grayscale by using the Kite *et al.* inverse halftoning algorithm [17]. The software *GNU Image Manipulation Program* (GIMP) is used to process the CE algorithm and to adjust the orientation of scanned images.

Fig. 3 shows the experimental results for the test images produced by the different methods. In this figure, the first column shows the original images (Fig. 3.(a)). The second, third and fourth columns show the restoration of color images with Queiroz's method and the PS process without CE (Fig. 3.(b)), with CE (Fig. 3.(c)), and without PS (Fig. 3.(d)), respectively. Likewise, the fifth, sixth, and seventh columns show the restorations using Ko's method with PS process without CE (Fig. 3.(e)), with CE (Fig. 3.(f)), and without PS (Fig. 3.(g)), respectively. The last column shows the restoration of the color image using the proposed method without CE or other post-processing techniques (Fig. 3.(h)).

As expected [40, 41], results obtained directly from the scanned images have faded colors, as depicted in Figs. 3 (b) and (e). The faded colors of these restored images can be intensified using a CE algorithm, as depicted in Figs. 3 (c) and (f). From these results, we can notice that Ko's method produces results with fewer color artifacts, but the colors are still different from the colors in the originals. In this case, color differences are not only due to the PS process, given that the colors of Figs. 3 (d) and (g) also differ from the original colors. For these methods, the color distortions are the consequence of the subsampling of the color channels that is performed during the encoding process (watermarking). On the other hand, in the proposed method the RGB channels are equally subsampled (from 24 bits per pixel to 3 bits per pixel) and a good inverse halftoning algorithm is used to restore the content [17]. Therefore, distortions are less prominent, as can be noticed from the results in Fig. 3 (h).

TABLE I
Objective evaluation of structural similarity between the original color watermarks and the restored watermarks using the tested methods (CMSSIM).

Image	Queiroz			Ko			Proposed
	PS	PS+CE	NoPS	PS	PS+CE	NoPS	
Airplane	0.561	0.315	0.456	0.600	0.323	0.336	0.734
Baboon	0.343	0.442	0.525	0.432	0.525	0.456	0.751
Girl	0.287	0.394	0.432	0.285	0.397	0.357	0.855
House	0.533	0.578	0.758	0.539	0.606	0.727	0.876
Lena	0.138	0.243	0.238	0.148	0.232	0.243	0.948
Peppers	0.340	0.625	0.782	0.339	0.683	0.794	0.933
Sailboat	0.390	0.529	0.803	0.490	0.594	0.782	0.825
Splash	0.334	0.628	0.830	0.256	0.479	0.854	0.901
Average	0.366	0.469	0.603	0.386	0.480	0.569	0.853

TABLE II
Objective evaluation of color fidelity between the original color watermarks and the restored watermarks using the tested methods (ΔE^*).

Image	Queiroz			Ko			Proposed
	PS	PS+CE	NoPS	PS	PS+CE	NoPS	
Airplane	09.95	17.29	17.02	12.02	21.67	21.35	07.52
Baboon	17.30	18.00	14.09	17.15	18.92	16.55	10.20
Girl	18.65	22.22	24.65	17.35	21.64	27.39	10.37
House	12.98	14.41	10.49	12.19	14.93	11.95	06.24
Lena	22.31	24.63	20.75	22.48	24.31	22.82	05.40
Peppers	22.01	18.22	15.63	21.05	16.42	17.15	05.54
Sailboat	17.52	16.59	11.54	16.97	17.47	13.37	08.58
Splash	20.62	18.66	9.52	22.23	23.37	09.27	06.47
Average	17.67	18.75	15.46	17.68	19.84	17.48	07.53

We use two objective quality metrics to quantitatively evaluate color distortions. The first metric is the Color Multi-scale Structural Similarity Index (CMSSIM) [47]. CMSSIM is used to measure the overall similarity between original and restored images. The second metric is the CIE color difference measure (ΔE^*) [48] that is used to measure the accuracy of the recovered colors. The higher the value of CMSSIM, the better the quality of the reconstructed image, while the smaller ΔE^* is the better the quality of the reconstructed image. In Tables I and II, the values of CMSSIM and ΔE^* are shown for all images. Notice that the proposed method presents the best performance, in accordance with the qualitative results presented in Fig. 3. The CMSSIM values shown in Table I suggest that the proposed method recovers the original image with high similarity of luminance, color, or both. The performance differences between proposed and conventional methods are more clearly shown by examining the ΔE^* values in Table II, which gives estimates of the perceptual color difference. Therefore, from these results, we can conclude

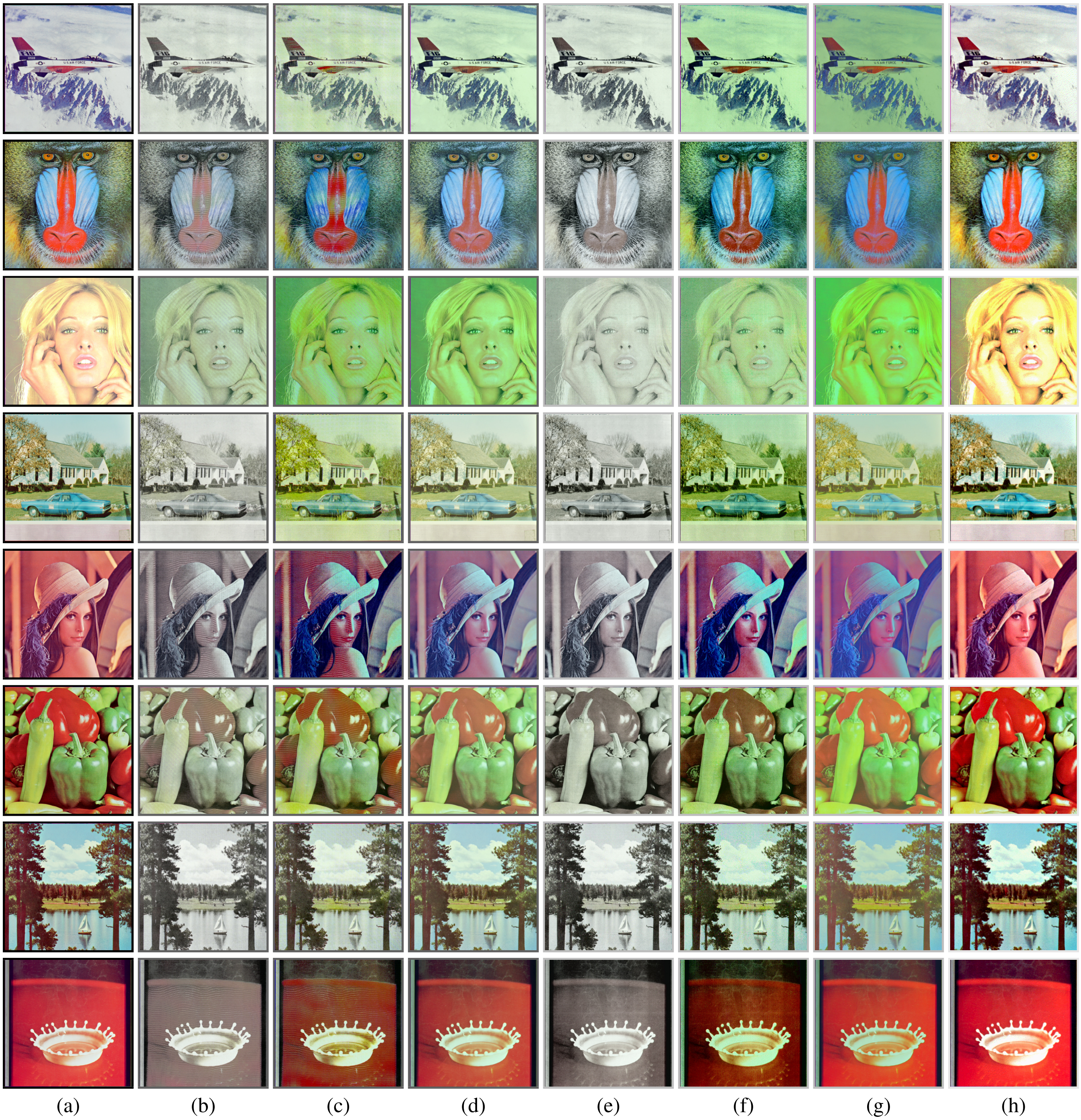

Fig. 3. Comparison of recovered color: (a) Original, (b) color image recovered from printed and scanned images using Queiroz's method without color enhancement, (c) color image recovered from printed and scanned images using Queiroz's method with color enhancement, (d) color image recovered from the textured grayscale image in simulation mode using Queiroz's method, (e) color image recovered from printed and scanned images using Ko's method without color enhancement, (f) color image recovered from printed and scanned images using Ko's method with color enhancement, (g) color image recovered from the textured grayscale image in simulation mode using Ko's method, and (h) color images recovered from halftone using the proposed method.

that the proposed method provides superior results for both structural and color fidelity.

B. Visual Impact of Halftoning Watermarking Algorithm

We tested the visibility of possible degradations caused by the embedding techniques using 14 dithering techniques. Halftone images generated with these techniques (without data embedding) are shown in Fig. 4. There has been relatively little research on the quality assessment of halftone images with the goal of better designing the printing and imaging system [55]. The use of image quality metrics to assess the quality of halftone images is not straightforward. Usually, image quality

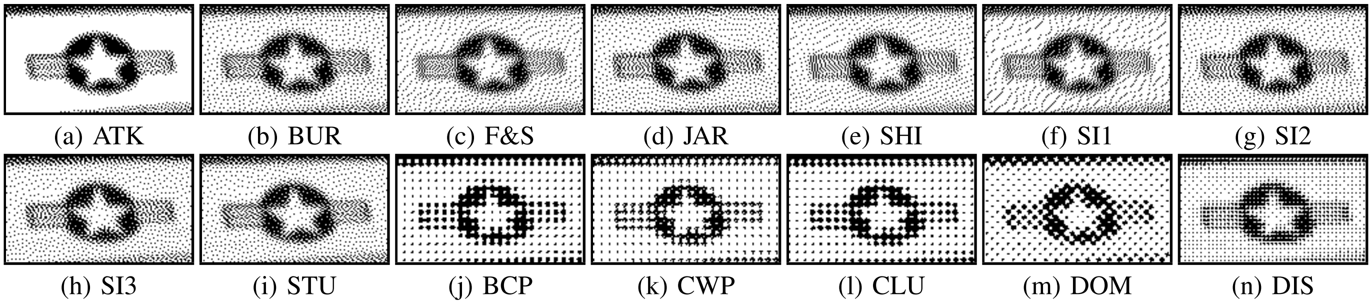

Fig. 4. Halftone versions of the ‘Airplane’ image computed using several algorithms. The error diffusion algorithms used are (a) Atkison [49], (b) Burkes [50], (c) Floyd-Steinberg [10], (d) Jarvis [8], (e) Shiau-Fan [51], (f)-(h) variants of Sierra’s algorithm [52], and (i) Stucki [9]. The ordered dithering algorithms used are (j) Balanced centered point, (k) Central white point, (l) Clustered dots, (m) Diagonal ordered matrix, and (n) dispersed dots [2, 53, 54].

metrics are designed to assess multi-level images (grayscale or color). However, these metrics are not effective to assess binary images (halftones), so we used the human visual peak signal to noise ratio (HPSNR) [12, 56] to assess the quality of halftone images. This metric is basically a weighted version of PSNR that incorporates the low-pass filtering characteristics of the HVS. This metric exploits the perceptual limitations of the HVS to determine if the dot patterns are being perceived as continuous gray levels [53].

TABLE III
HPSNR VALUES COMPARING MARKED HALFTONE AND ORIGINAL GRAYSCALE IMAGE.

Image	Dice, Jaccard, Sokal-Sneath	Matching, Rogers, Sokal-Michener	Russell	Yule
Airplane	43.74401	39.35379	39.83867	11.10960
Baboon	43.36772	35.89738	38.78762	14.51172
Girl	40.99878	42.46510	38.45246	10.41496
House	42.37456	39.30285	37.68586	12.28634
Lena	40.31220	40.98477	35.94697	16.46699
Peppers	42.80871	36.12626	40.95991	14.55234
Sailboat	42.63261	38.18374	39.35358	13.98017
Splash	43.88915	38.05749	41.10926	17.71411
Average	42.51597	38.79642	39.01679	13.87953

We tested the proposed method using the following dissimilarity measures: Dice, Jaccard-Needham, Matching, Rogers-Tanimoto, Russell-Rao, Sokal-Michener, Sokal-Sneath, and Yule. These dissimilarity measures are detailed by Zhang and Srihari [44]. We use these measures as the comparison criteria (D) for the minimization problem of Eq. 4. Table III shows the HPSNR of the test images, corresponding to the halftone visual quality.

From Table III, we observe that the same HPSNR values were obtained for the Dice, Jaccard-Needham, and Sokal-Sneath measures. In other words, although these measures produce distinct numbers, the selected mask is the same and, therefore, the watermarked halftone is exactly the same when we use these 3 measures. Similarly, the Matching, Russell-Rao, and Sokal-Michener measures produce the same HPSNR results. From all tested measures, Dice, Jaccard-Needham, and Sokal-Sneath are the most suitable as selection criteria, given that all watermarked halftones generated with these measures have HPSNR above 40dB. These results suggest that the watermarked halftones are visually acceptable and have a quality comparable to the quality of the corresponding original grayscale image.

Fig. 5 shows the visual difference produced by distinct dissimilarity measures for the ‘Lena’ image. Fig. 5 (a) shows the unmarked grayscale image and Fig. 5 (b) shows the grayscale converted to halftone using Floyd-Steinberg [10] (no watermarking). Figs. 5 (c), (d), (e), and (f) show the result of embedding color watermarks to Fig. 5 (b) using Dice, Matching, Russel, and Yule dissimilarity measures, respectively.

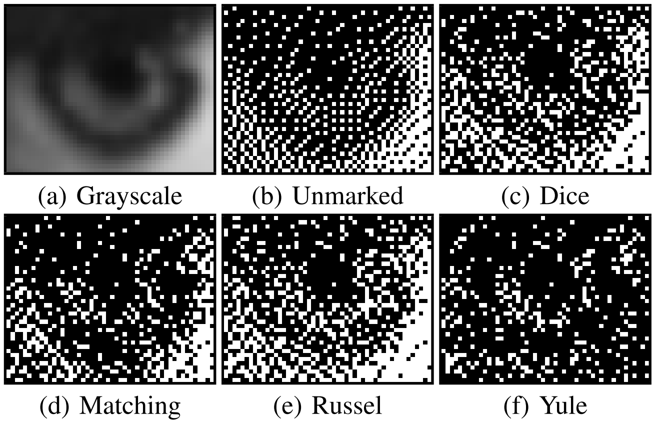

Fig. 5. Effect of dissimilarity measure on marked halftones.

Fig. 6 shows the result of a pairwise comparison using the tested halftoning algorithms for the ‘Lena’ image. In this figure, the horizontal axis corresponds to the algorithm used to compute the host halftone image, while the vertical axis corresponds to the algorithm used to compute the color channel halftone. The darker blocks represent the combinations that provide higher HPSNR values. From this figure, we observe that the dithering algorithm used to compute the halftone of the host has a bigger influence on quality than the dithering algorithms used to compute the halftone of the watermark color channels. Moreover, the quality of the coded halftone is higher when Atkinson’s algorithm is used. This pairwise comparison was performed for all tested images and results are very similar. Therefore, the performance of the dithering combination does not seem to be affected by image content.

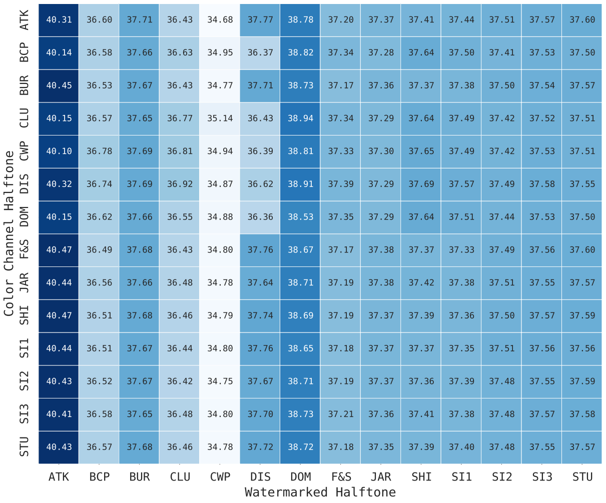

Fig. 6. Pairwise comparison between the dithering algorithm used on host halftone image and the dihering algorithm used to compute the binary version of RGB channels of embedded watermark. In this case, the halftone of 'Lena' image is used as both host and embedded image.

C. Computational Cost

Halftoning and inverse halftoning algorithms are usually implemented on embedded hardware, where clock speeds and memory sizes tend to be lower than in desktop-class machines. Consequently, the number of operations required to execute the algorithm and the amount of memory consumed are important issues. Using the 'Dice' distance measure as the optimization criterion and the Floyd-Steinberg as the halftoning algorithm, we evaluated the computational cost of the proposed method. We performed all simulations under the same conditions. Color enhancement was not considered in this test. For each image, we performed the simulation 10 times, collecting the encoding and decoding running time and, at the end, calculating the average times. This avoids the bias caused by other processes running at the same time on the computer.

TABLE IV
RUNTIMES OF TESTED ALGORITHMS (IN SECONDS).

Image	Queiroz		Ko		Proposed	
	Encode	Decode	Encode	Decode	Encode	Decode
Airplane	2.871	3.251	2.778	2.823	288.006	0.738
Baboon	2.933	3.211	2.823	2.806	279.809	0.698
Girl	2.861	3.187	2.812	2.955	270.514	0.687
House	2.851	3.195	2.865	2.832	269.090	0.695
Lena	2.848	3.234	2.768	2.924	268.655	0.700
Peppers	2.821	3.180	2.783	2.857	269.939	0.695
Sailboat	2.840	3.277	2.897	2.882	282.386	0.797
Splash	2.862	3.193	2.766	2.807	337.060	0.781
Average	2.861	3.216	2.812	2.861	283.182	0.724

Table IV shows the average elapsed time obtained for the proposed method and two other algorithms (Queiroz and Ko). Notice that the algorithms by Queiroz and Ko have similar encoding running times. The proposed algorithm has a much higher encoding time, with the optimization problem being responsible for a big percentage of this time (see Eq. 4). The optimization problem can be solved in approximately 0.001s per pixel, resulting in an average of 283.182s for an image of 512×512 pixels. Although the encoding time of the proposed algorithm is higher than for the other two methods, the decoding time is faster because the triplets are recovered using a simple hash table. After recovering the triplets and reconstructing the halftone color channels, the inverse halftone must be performed three times. But, since the chosen inverse halftoning algorithm is fast, the decoding time is tiny.

V. CONCLUSIONS

We have presented a method for directly embedding and recovering a color image watermark into a halftone image. This method challenges previous works because it enables embedding a color image into a binary (black-and-white) image. It gives the halftone channel a higher capacity. Experimental results show that the recovered colors have higher fidelity than that obtained using 'color-to-grayscale and back' methods. In addition, the proposed method is not limited to color reconstruction. Since the triplets containing the RGB information are embedded and recovered from the masked halftone, we can use the proposed approach to embed any color content into a binary host image (halftone). The proposed method can be used in applications, such as steganography, hardcopy data hiding, etc.

Although the quality of the colors restored with the proposed method is superior, the required processing time is significantly higher. To reduce the computation time, two approaches can be investigated in the future. Firstly, the implemented mask construction process is not fully optimized. Therefore, we can use machine learning algorithms to obtain an efficient distribution of the mask groups. With optimized mask groups, redundant processing can be avoided and the minimization problem can be simplified. Secondly, the proposed method can be parallelized in order to be used in real-time applications.

REFERENCES

[1] D. Muselet and A. Trémeau, "Recent trends in color image watermarking," *Journal of Imaging Science and Technology*, vol. 53, no. 1, pp. 10 201–1, 2009.

[2] D. E. Knuth, "Digital halftones by dot diffusion," *ACM Transactions on Graphics (TOG)*, vol. 6, no. 4, pp. 245–273, 1987.

[3] J. S. Viggiano, "Modeling the color of multi-colored halftones," in *Proc. TAGA*, vol. 42, 1990, pp. 44–62.

[4] D. J. Lieberman and J. P. Allebach, "Efficient model based halftoning using direct binary search," in *Image Processing, 1997. Proceedings., International Conference on*, vol. 1. IEEE, 1997, pp. 775–778.

[5] S. H. Kim and J. P. Allebach, "Impact of hvs models on model-based halftoning," *Image Processing, IEEE Transactions on*, vol. 11, no. 3, pp. 258–269, 2002.

[6] R. Ulichney, "The void-and-cluster method for dither array generation," *SPIE MILESTONE SERIES MS*, vol. 154, pp. 183–194, 1999.

[7] X. Liu, Y. Geng, and Z.-J. Li, "Compression method for ordered dither halftone image [j]," *Journal of Computer Applications*, vol. 1, p. 039, 2011.

[8] J. F. Jarvis, C. N. Judice, and W. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," *Computer Graphics and Image Processing*, vol. 5, no. 1, pp. 13–40, 1976.

[9] P. Stucki, *MECCA: a multiple-error correction computation algorithm for bi-level image hardcopy reproduction.* IBM Thomas J. Watson Research Division, 1981.

[10] R. W. Floyd, "An adaptive algorithm for spatial gray-scale," in *Proc. Soc. Inf. Disp.*, vol. 17, 1976, pp. 75–77.

[11] C. Schmaltz, P. Gwosdek, A. Bruhn, and J. Weickert, "Electrostatic halftoning," in *Computer Graphics Forum*, vol. 29, no. 8. Wiley Online Library, 2010, pp. 2313–2327.

[12] J.-M. Guo and Y.-F. Liu, "Improved block truncation coding using optimized dot diffusion," *Image Processing, IEEE Transactions on*, vol. 23, no. 3, pp. 1269–1275, 2014.

[13] P. Gwosdek, C. Schmaltz, J. Weickert, and T. Teuber, "Fast electrostatic halftoning," *Journal of Real-Time Image Processing*, pp. 1–14, 2014.

[14] M. Mese and P. P. Vaidyanathan, "Look-up table (lut) method for inverse halftoning," *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1566–1578, 2001.

[15] Y.-F. Liu and J.-M. Guo, "New class tiling design for dot-diffused halftoning," *Image Processing, IEEE Transactions on*, vol. 22, no. 3, pp. 1199–1208, 2013.

[16] P. G. Freitas, M. C. Farias, and A. P. de Araújo, "Fast inverse halftoning algorithm for ordered dithered images," in *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*. IEEE, 2011, pp. 250–257.

[17] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, "A fast, high-quality inverse halftoning algorithm for error diffused halftones," *Image Processing, IEEE Transactions on*, vol. 9, no. 9, pp. 1583–1592, 2000.

[18] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[19] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.

[20] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *Image Processing, IEEE Transactions on*, vol. 6, no. 12, pp. 1673–1687, 1997.

[21] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1181–1196, 1999.

[22] L. Tan, X. Sun, and G. Sun, "Print-scan resilient text image watermarking based on stroke direction modulation for chinese document authentication," *Radioengineering*, vol. 21, no. 1, pp. 170–181, 2012.

[23] B. Yang, X. Sun, X. Chen, J. Zhang, and X. Li, "An efficient forensic method for copy–move forgery detection based on dwt-fwht," *Radioengineering*, vol. 22, no. 4, 2013.

[24] M. Chroni, A. Fylakis, and S. D. Nikolopoulos, "Watermarking images in the frequency domain by exploiting self-inverting permutations," 2013.

[25] R. Todmal Satish and K. Thammi Reddy, "A technique to find optimal location for wavelet-based image watermarking using genetic algorithm," *Machine graphics & vision*, vol. 20, no. 2, pp. 173–196, 2011.

[26] A. Umamageswari and G. R Suresh, "A new cryptographic digital signature for secure medical image communication in telemedicine," *International Journal of Computer Applications*, vol. 86, no. 11, pp. 4–9, 2014.

[27] Q. Hou, D. Junping, L. Li, J. Lu, and C.-C. Chang, "Scanned binary image watermarking based on additive model and sampling," *Multimedia Tools and Applications*, pp. 1–20.

[28] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *Multimedia, IEEE Transactions on*, vol. 6, no. 4, pp. 528–538, 2004.

[29] L. Zongqing and Z. Hongbin, "A new data hiding method in binary images," in *Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on*, vol. 3. IEEE, 2006, pp. 66–69.

[30] F. Daraee and S. Mozaffari, "Watermarking in farsi/arabic binary document images using fractal coding," in *Information Security and Cryptology (ISCISC), 2011 8th International ISC Conference on*. IEEE, 2011, pp. 67–72.

[31] B. He, Y. Wu, K. Kang, and W. Guo, "A robust binary text digital watermarking algorithm for print-scan process," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 7. IEEE, 2009, pp. 290–294.

[32] H.-C. Wu, Y.-C. Liu, K.-C. Liu, and C.-S. Tsai, "A block-based authentication watermarking technique for binary images," in *Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on*, vol. 3. IEEE, 2008, pp. 513–518.

[33] J.-S. Pan, H. Luo, and Z.-M. Lu, "A lossless watermarking scheme for halftone image authentication," *International Journal of Computer Science and Network Security*, vol. 6, no. 2b, pp. 147–151, 2006.

[34] J.-M. Guo and Y.-F. Liu, "High capacity data hiding for error-diffused block truncation coding," *Image Processing, IEEE Transactions on*, vol. 21, no. 12, pp. 4808–4818, 2012.

[35] C.-H. Son and H. Choo, "Watermark detection from clustered halftone dots via learned dictionary," *Signal Processing*, vol. 102, pp. 77–84, 2014.

[36] J.-M. Guo, C.-C. Su, Y.-F. Liu, H. Lee, and J.-D. Lee, "Oriented modulation for watermarking in direct binary search halftone images," *Image Processing, IEEE Transactions on*, vol. 21, no. 9, pp. 4117–4127, 2012.

[37] J.-M. Guo and Y.-F. Liu, "Hiding multitone watermarks in halftone images," *IEEE MultiMedia*, no. 1, pp. 34–43, 2010.

[38] C.-H. Son and H. Choo, "Color recovery of black-and-white halftoned images via categorized color-embedding look-up tables," *Digital Signal Processing*, vol. 28, pp. 93–105, 2014.

[39] C.-H. Son, K. Lee, and H. Choo, "Inverse color to black-and-white halftone conversion via dictionary learning and color mapping," *Information Sciences*, vol. 299, pp. 1–19, 2015.

[40] R. L. de Queiroz, "Reversible color-to-gray mapping using subband domain texturization," *Pattern Recognition Letters*, vol. 31, no. 4, pp. 269–276, 2010.

[41] K.-W. Ko, O.-S. Kwon, C.-H. Son, and Y.-H. Ha, "Color embedding and recovery based on wavelet packet transform," *Journal of Imaging Science and Technology*, vol. 52, no. 1, pp. 10 501–1, 2008.

[42] R. L. de Queiroz and K. M. Braun, "Color to gray and back: color embedding into textured gray images," *Image Processing, IEEE transactions on*, vol. 15, no. 6, pp. 1464–1470, 2006.

[43] H. Kekre, S. D. Thepade, and R. N. Chaturvedi, ""color to gray and back" using dst-dct, haar-dct, walsh-dct, hartley-dct, slant-dct, kekre-dct hybrid wavelet transforms," in *Proceedings of the Third International Conference on Soft Computing for Problem Solving*. Springer, 2014, pp. 613–623.

[44] B. Zhang and S. N. Srihari, "Binary vector dissimilarity measures for handwriting identification," in *Proc. of SPIE Vol*, vol. 5010, 2001, p. 29.

[45] ——, "Properties of binary vector dissimilarity measures," in *Proc. JCIS Int'l Conf. Computer Vision, Pattern Recognition, and Image Processing*, vol. 1, 2003.

[46] A. G. Weber, "The USC-SIPI image database version 5," *USC-SIPI Report*, vol. 315, pp. 1–24, 1997.

[47] M. Hassan and C. Bhagvati, "Structural similarity measure for color images," *International Journal of Computer Applications*, vol. 43, no. 14, pp. 7–12, 2012.

[48] G. Sharma, W. Wu, and E. N. Dalal, "The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research & Application*, vol. 30, no. 1, pp. 21–30, 2005.

[49] B. Atkinson, "correspondence with john balestrieri, january 2003 (unpublished)."

[50] D. Burkes, "Presentation of the burkes error filter for use in preparing continuous-tone images for presentation on bi-level devices, in lib 15, cis graphics support forum (unpublished)," 1988.

[51] J.-N. Shiau and Z. Fan, "Set of easily implementable coefficients in error diffusion with reduced worm artifacts," pp. 222–225, 1996. [Online]. Available: http://dx.doi.org/10.1117/12.236968

[52] F. Sierra, "in lib 17 (developer's den), cis graphics suppor forum (unpublished)."

[53] R. Ulichney, *Digital halftoning*. MIT press, 1987.

[54] B. E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures," *SPIE MILESTONE SERIES MS*, vol. 154, pp. 139–143, 1999.

[55] P. Itoua, A. Beghdadi *et al.*, "Objective perceptual evaluation of halftoning using image quality metrics," in *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*. IEEE, 2010, pp. 456–459.

[56] J.-M. Guo and Y.-F. Liu, "Joint compression/watermarking scheme using majority-parity guidance and halftoning-based block truncation coding," *Image Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 2056–2069, 2010.