IMPROVED PERFORMANCE OF INVERSE HALFTONING ALGORITHMS VIA COUPLED DICTIONARIES

Pedro García Freitas*, Mylène C.Q. Farias**, and Ailetéia P. F. de Araújo*

*Department of Computer Science, †Department of Electrical Engineering
University of Brasilia, Brasília, Brazil

ABSTRACT

Inverse halftoning techniques are known to introduce visible distortions (typically, blurring or noise) into the reconstructed image. To reduce the severity of these distortions, we propose a novel training approach for inverse halftoning algorithms. The proposed technique uses a coupled dictionary (CD) to match distorted and original images via a sparse representation. This technique enforces similarities of sparse representations between distorted and non-distorted images. Results show that the proposed technique can improve the performance of different inverse halftone approaches. Images reconstructed with the proposed approach have a higher quality, showing less blur, noise, and chromatic aberrations.

Index Terms— Inverse Halftone, Dictionary Training, Image Restoration, Sparse Coding, Sparse Modeling.

1. INTRODUCTION

Most printed materials are produced using halftoning techniques. Halftoning is the technique of converting continuous-tone images into images with a limited number of color levels. The technique generates images that, although having a limited number of levels, convey the illusion of having a higher number of levels. Inverse halftoning is the process of generating continuous-tone images from their halftoned versions. The reconstruction of scanned images is an application of inverse halftoning that is very important for the publishing industry [1]. Other applications include the protection of digital documents against piracy [2], authentication of video content [3], compression of images [4], and error concealment for images and videos [5]. In all these applications, the quality of the reconstructed image using the inverse halftoning algorithm is crucial.

Generally, algorithms proposed for inverse halftoning focus on the restoration of a specific kind of halftoning algorithm. In the case of dithering, there are several techniques to generate the patterns that create the illusion of a continuous-tone image. The corresponding inverse halftoning algorithms use the appropriate pattern to optimize image reconstruction. For example, Saika et. al. [6] and Freitas et al. [7] use stochastic models to restore continuous-tone image from Ordered Dithering (OD) halftones. But, when the halftoning technique is an error diffusion dithering technique, these approaches do not produce good results. This is a problem since error diffusion techniques [8] have a better performance than ordered dithering techniques [9, 10].

In this paper, we present a method to enhance the visual quality of images reconstructed using inverse halftoning techniques. We treat the inverse halftoned image as a distorted version of the original image and focus on recovering a non-distorted version. Although the technique presented here can be used to obtain the inverse halftone directly, the goal of the proposed technique is to detect and reduce distortions in reconstructed images generated by an inverse halftoning technique.

This paper is organized as follows. In Section 2, we describe the distortions associated with inverse halftoning algorithms. In Section 3, we discuss how to train a pair of dictionaries to match information between the original image and the reconstructed image. Section 4 details our strategy to improve inverse halftoning methods and Section 5 presents its results. Finally, in Section 6, we present our conclusions and discuss future works.

2. INVERSE HALFTONING

In this work, halftoning is the process of generating a binary (2 levels) image \( I_b \) from a continuous-tone (255-levels) grayscale image \( I_g \) (or a channel of a colored image). Particularly,

\[
I_b = H \cdot I_g, \tag{1}
\]

where \( H \) is an operator representing the halftoning process that transforms \( I_g \) into the binary image (pixels values equal to ‘1’ or ‘0’). Although \( H \) can be viewed as a simple operation, there are several ways to model it. As presented by Ulichney [11], different models correspond to different dithering patterns, leading to different ways to correlate the information between \( I_b \) and \( I_g \).

Inverse halftoning is the process of reconstructing \( I_g \) from \( I_b \). Since to generate \( I_b \) a considerable amount of information is discarded, the inverse halftoning algorithm can only generate an approximation of \( I_g \). In other words, the reconstructed
image $\hat{I}_g$ contains an error, as modeled by following equation:

$$I_g = \hat{I}_g + \epsilon,$$  

(2)

where $\epsilon$ is an approximation error and $\hat{I}_g$ is the result of inverse halftoning process.

This approximation error $\epsilon$ is characterized by a degradation of the reconstructed image that we call distortion. The visual effect of this distortion varies according to the inverse halftoning technique, as shown in Fig. 1. The simplest inverse halftoning algorithm consists of low-pass filtering a halftoned image. This approach removes the noise introduced by halftoning patterns, but it also removes high-frequency information. Comparing the original Lena image (Fig. 1 (a)) and the reconstructed version generated using Wavelet-based inverse halftoning via deconvolution (WInHD) [10] (Fig. 1 (e)), we notice that the reconstructed image is blurred. Comparing the original (Fig. 1 (a)) and the image reconstructed using Fast Inverse Halftoning Algorithm for Ordered Dithered Images (FIHT) [7] (Fig. 1 (g)), we notice that the reconstructed image is noisier.

To tackle this problem, we observe the sparse representation of signals [12] and notice that there are linear relationships among non-distorted and distorted signals. Therefore, it is possible to reconstruct an original image from its distorted version [13, 14]. To enhance the quality of reconstructed images, in this work we use a sparse representation of the original and halftoned images and an error concealment technique based on error correcting codes. The mathematical model used for the sparse representation is based on dictionary learning [15, 16], which is a matrix factorization problem that finds a dictionary (usually overcomplete) that sparsely encodes the fitted data. This work is inspired by the approach proposed by Yang et al. [17] that reduces blurring in interpolated images. Our work proposes a general model that reduces distortions in continuous-tone images reconstructed from halftoned images.

3. DICTIONARY LEARNING

Sparse representations make it possible to approximate a given signal, $x \in \mathbb{R}^d$, by a linear combination of elementary sets. These elementary sets are called basic atoms and can be stored in an over-complete dictionary $D \in \mathbb{R}^{d \times k}$, where $d \ll k$. Sparse coding techniques consist of discover-
ing a good set of basic atoms to represent $D$, what is usually done by learning from a set of training samples composed by patches of the signal $x$, i.e. $x = \{x_1, x_2, \ldots, x_n\}$.

In the literature, there are many approaches to find dictionaries that guarantee the recovery of signals using sparse representations [16, 15, 18, 19]. In this paper, we use a learning-dictionary strategy that minimizes the differences between the signal and its sparse representation. In other words, we use the following optimization problem:

$$\arg \min_{D, \alpha} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

subject to $\|D_i\|_2^2 \leq 1 \quad i = 1, 2, \ldots, m,$

where $D_i$ is the $i$-th column of $D$, $\lambda$ is a constant that multiplies the $\ell_1$ term (regularization parameter), and $\alpha_i$ is the sparse representation of $x_i$. This problem is convex when $D$ or $\{\alpha_i\}$ are fixed. When $D$ is fixed, $\{\alpha_i\}$ can be solved efficiently by linear programming, like in the Lasso problem [20, 21]. Otherwise, by fixing $\{\alpha_i\}$, $D$ can be solved as a constrained quadratic problem [22]. This joint optimization problem converges to a local minimum [23].

3.1. Coupled Dictionary Learning

Although the strategy described in the previous section is efficient to represent a signal $x$, we need a strategy to find the relationship between distorted and non-distorted signals. Our approach consists of learning two dictionaries, $D_x$ and $D_y$, and find the relationship between these two spaces.

For this, the distorted signal $x$ in terms of $D_x$ is the same as the non-distorted signal $y$ in terms of $D_y$. A formulation to this problem is made by generalizing the single dictionary sparse coding as follows [17]:

$$\arg \min_{D_x, D_y, \alpha} \|x_i - D_x\alpha_i\|_2^2 + \|y_i - D_y\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

subject to $\|D_x\|_2^2 \leq 1$

subject to $\|D_y\|_2^2 \leq 1$

where

$$D_k = \arg \min_{D_k, \alpha} \|k_i - D_k\alpha_k\|_2^2 + \lambda \|\alpha_k\|_1,$$

for $k \in \{x, y\}$. We chose $\alpha^x_i = \alpha^y_i$ to guarantee that the distorted and non-distorted signals share the same sparse representation $\alpha_i$.

Combining Equations 4 and 5, we can rewrite the optimization problem as:

$$\min_{D_x, \alpha} \|z_i - D_x\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

subject to $\|D_x\|_2^2 \leq 1$,

where

$$z_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

and $D_c$ is the coupled dictionary, given by:

$$D_c = \begin{bmatrix} D_x \\ D_y \end{bmatrix}.$$

4. PROPOSED METHOD

In this section, we discuss how to reduce distortions of images reconstructed with inverse-halftoning algorithms using the Coupled Dictionaries approach described in the previous section. For this approach, the distorted ($y$) and original ($x$) images are decomposed into sets of patches, i.e. $x = \{x_1, x_2, \ldots, x_n\}$ and $y = \{y_1, y_2, \ldots, y_n\}$, respectively. The main goal is to map the two spaces using coupled sparse dictionaries and, then, use these dictionaries to recover a patch of $x_i$ from a patch of $y_i$ ($i = 1, 2, \ldots, n$).

4.1. Learning Stage

The first stage of the proposed method consists of extracting distorted and non-distorted patch-pairs from the distorted and non-distorted images, respectively. We use a database of training images, which is composed of a set of original (gray-scale) images and a set of corresponding images reconstructed using inverse halftoning algorithms. To generate the database, we first pick $N$ training non-distorted images $\{I^o_g\}_{n=1}^N$.

Then, we generate their $N$ halftone versions $\{I^n_g\}_{n=1}^N$. Finally, for each halftoned image, we use an inverse halftoning algorithm to obtain each distorted image $\{I^n_g\}_{n=1}^N$.

With these pairs of original and reconstructed images, $I^o_g$ and $I^n_g$, we compute a random permutation of correspondent $8 \times 8$ patches blocks, generating a set of original and reconstructed pairs of blocks. The pixels in the $i$-th pair of blocks, $\{b^o_n, b^n_n\}$, are composed by the original information and its corresponding reconstructed information, respectively. These blocks are resized to vectors and referred as patch pairs. Patch pairs are appended to a list $p_i$ in which the $i$-th element $p_i$ is the previously defined pair $(x_i, y_i)$, where $x = \{x_1, x_2, \ldots, x_n\}$ is the set of patches from original images (non-distorted) and $y = \{y_1, y_2, \ldots, y_n\}$ is the corresponding set from reconstructed images (distorted).

Fig. 3 depicts the steps used for the extraction of the patch pairs. In step (1), a single image is converted to a halftoned image. In step (2), this halftoned image is reconstructed using an inverse halftoning technique. Then, in steps (3) and (4), the distorted and original images are divided into blocks. In step (5), the original image block and its corresponding block in the reconstructed image are extracted, generating the pair $(b_i, b_i)$. Finally, in step (6), the non-distorted and distorted blocks are reshaped into vectors to create the patch pair $p_i = (x_i, y_i)$.

To learn dictionaries from distorted and non-distorted patches, we have to keep the sparse representation of non-distorted patches similar to the sparse representation of distor-
Fig. 3. First stage of proposed technique: extraction of patch pairs to create dictionaries.

ted patches. As mentioned earlier, non-distorted and distorted patches share the same \( \{ \alpha_i \} \), i.e. \( \alpha_i^x = \alpha_i^y = \alpha_i \) in Equation 5. In other words, for representing distorted and original information, the dictionaries are coupled by sharing the same sparse representation.

4.2. Minimizing Distortions Stage

Once the training is done, we have two dictionaries, \( D_x \) and \( D_y \), corresponding to non-distorted and distorted patches, respectively. To reduce the distortion of reconstructed images, we first extract the distorted patches \( \{ y_i \} \) from \( \hat{I}_g \). Then, for each \( i \)-th distorted patch, we find the sparse representation corresponding to \( D_y \), using the following equations:

\[
\arg \min_{\alpha_i} \left\| \alpha_i \right\|_1 \quad \text{subject to} \quad \left\| D_y \alpha_i - y_i \right\|_2^2 \leq \epsilon. \tag{9}
\]

This problem can be solved using one of several linear regression statistical techniques for \( \ell_1 \)-norm [21, 23]. For a given optimal solution \( \alpha_i \), the corresponding non-distorted patch is given by:

\[
x_i = D_x \alpha_i. \tag{10}
\]

After non-distorted patches \( \{ x_i \} \) are obtained from the distorted ones \( \{ y_i \} \), we generate the reconstructed image \( \hat{I}_g \) by substituting each non-distorted patch \( \{ x_i \} \) into the corresponding position in the binary image. In other words,

\[
\hat{I}_g = I_g + \sigma, \tag{11}
\]

where \( \sigma < \epsilon \) (see Equation 2) and, therefore, \( \hat{I}_g \) is closer to \( I_g \) than \( \hat{I}_g \).

5. EXPERIMENTAL RESULTS

We used a set of 64 images to train the dictionaries. These images were extracted from the “Texture” set of USC-SIPI Image Database [24]. Fig. 4 shows a sample of these training images. The texture images varied from coarse to fine and from smooth to busy. Notice that the training images have a simple content, consistent with what is frequently found in blocks of larger and complex images.

To generate the distorted images, we used two different types of inverse halftoning techniques: FIHT [7] and WinHD [10]. These techniques were chosen because these algorithms use different approaches to reconstruct continuous-tone images. While WinHD reconstructs images from error diffusion halftones, FIHT restores images from ordered dithered halftones. As a consequence, the visual distortions produced by each algorithm look very different, as can be seen in Figs. 1 (e) and (g). Therefore, this choice of techniques makes it possible to test the proposed method for two very different halftoning algorithms. Both methods were implemented in Matlab® and the simulations were performed in a laptop with an Intel i7-4700MQ processor and 32GB of RAM.

Fig. 4. Example of images used in the training stage [24].

Fig. 5. Images used in experimental tests. In order: Airplane, Baboon, Girl, House, Lena, Peppers, Sailboat, and Splash.

The proposed approach is tested using a set of 8 color natural images extracted from the “Miscellaneous” set of USC-SIPI Image Database [24]. These images are depicted in Fig. 5. For each image, to test the performance of the proposed technique for WinHD and FIHT, we generate an error diffusion halftoned image and an ordered dithered halftoned image, respectively. Since the images are colored, we generate two halftoned images for each RGB channel.

The results obtained for a detail of the image ‘Lena’ are shown in Fig. 1. Fig. 1 (a) corresponds to the original image, while Figs. 1 (b) and (c) correspond to its halftoned versions obtained using the Jarvis’s error diffusion method and
Table 1. SSIM values for images reconstructed with inverse halftoning algorithms.

<table>
<thead>
<tr>
<th>Image</th>
<th>WInHD</th>
<th>WInHD+CD</th>
<th>FIHT</th>
<th>FIHT+CD</th>
<th>CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>0.91833</td>
<td>0.92137</td>
<td>0.88987</td>
<td>0.91487</td>
<td>0.51259</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.76191</td>
<td>0.76934</td>
<td>0.83559</td>
<td>0.85362</td>
<td>0.57378</td>
</tr>
<tr>
<td>Girl</td>
<td>0.89191</td>
<td>0.89611</td>
<td>0.89713</td>
<td>0.92071</td>
<td>0.50451</td>
</tr>
<tr>
<td>House</td>
<td>0.88623</td>
<td>0.89091</td>
<td>0.86406</td>
<td>0.88107</td>
<td>0.41437</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.85510</td>
<td>0.85742</td>
<td>0.84406</td>
<td>0.86044</td>
<td>0.47974</td>
</tr>
<tr>
<td>Sailboat</td>
<td>0.84204</td>
<td>0.84621</td>
<td>0.83499</td>
<td>0.85400</td>
<td>0.55574</td>
</tr>
<tr>
<td>Splash</td>
<td>0.89055</td>
<td>0.89578</td>
<td>0.85100</td>
<td>0.86576</td>
<td>0.48803</td>
</tr>
</tbody>
</table>

As mentioned earlier, comparing Figs. 1 (a) and (e), we observe that the restoration using WInHD produces a blurred distortion. This distortion is concealed when we use the proposed technique (WInHD+CD), as shown in Fig. 1 (f). On the other hand, FIHT produces a sharper and noisier image, as shown in Fig. 1 (g). The noise is minimized using the proposed technique (FIHT+CD), as depicted in Fig. 1 (h).

Fig. 1 (d) shows the results of using the proposed technique directly on the image shown in Fig. 1 (b). This example shows that the training stage can be performed without using the inverse halftoning algorithm. Therefore, the proposed method can be used to restore graylevels directly from binary images, i.e. the proposed method can act as an inverse halftoning algorithm too.

Fig. 2 shows the results for a detail of the ‘Airplane’ image. For better visualization of the distortions, a digital zoom was applied to the details of the image. When we compare the images in Figs. 2 (a), (e), and (f), we can notice that distortions in high-frequency content are minimized. Moreover, the noisy distortion inserted by FIHT method is also minimized by CD, as can be noticed when comparing Fig. 2 (a), (g), and (h).

Table 1 shows estimates of the quality of reconstructed images obtained using a popular full-reference image quality metric: the structural similarity index (SSIM) [25]. The second column of the table shows the SSIM values corresponding to the images obtained using the WInHD inverse halftone process, while the third column shows the SSIM values obtained using the proposed method for WInHD halftoned images. Similarly, the fourth and fifth column show the same results for the FIHT method. The last column shows the SSIM values corresponding to the images reconstructed using the CD directly on Jarvis’ halftoned images. We can notice that the use of CD increases the SSIM values of images reconstructed for WInHD and FIHT halftoning techniques.

The scalability of algorithm is analysed by varying the number of patches used in the training stage. As shown in Fig. 6, we can observe that the training time increases linearly with the number of samples. Moreover, the sampling time is approximately constant (130 seconds average) and much smaller than the training time. This indicates that the main computational cost is, as expected, the optimization process. Therefore, an optimized implementation of the Lasso algorithm is required for implementation of the proposed algorithm on embedded hardware (e.g. scanning devices) or for real-time applications.

6. CONCLUSIONS

We have presented a novel approach to improve the quality of images reconstructed using inverse halftoning algorithms. The proposed approach uses coupled dictionaries, which are trained using original images and images reconstructed using inverse halftoning algorithms. Experimental results show that the proposed method is able to conceal distortions caused by inverse halftoning techniques, improving the quality of images generated by these techniques. Moreover, we have shown that coupled dictionaries can also be used as an inverse halftoning method. Future works include the investigation of the proposed method performance gain in inverse halftone applications, such as error concealment, image inpainting, digital content protection, and digital scanning software. Also, a study of the model’s optimal parameter values is needed. Finally, further investigation of the effect of the training image database on the quality of the restored images is necessary.

7. REFERENCES


