Abstract—Currently most electrophotographic printers use halftoning technique to print continuous tone images, so scanned images obtained from such hard copies are usually corrupted by screen like artifacts. In this paper, a new model of scanned halftone image is proposed to consider both printing distortions and halftone patterns. Based on this model, an adaptive filtering based descreening method is proposed to recover high quality contone images from the scanned images. Image redundancy based denoising algorithm is first adopted to reduce printing noise and attenuate distortions. Then screen frequency of the scanned image and local gradient features are used for adaptive filtering. Basic contone estimate is obtained by filtering the denoised scanned image with an anisotropic Gaussian kernel, whose parameters are automatically adjusted with the screen frequency and local gradient information. Finally, an edge-preserving filter is used to further enhance the sharpness of edges to recover a high quality contone image. Experiments on real scanned images demonstrate that the proposed method can recover high quality contone images from the scanned images. Compared with the state-of-the-art methods, the proposed method produces very sharp edges and much cleaner smooth regions.

Index Terms—Scanned image, descreening, inverse halftoning, adaptive filtering, steerable filter.

I. INTRODUCTION

In order to print continuous tone (contone) images, electrophotographic (EP) printers adopt halftone technique to convert contone image into bilevel halftone image. When one bilevel halftone image is printed on paper, its hard copy is comprised of halftone patterns, which are hardly perceived by human eyes. However, when the hard copy printed by EP printer is scanned, annoying artifacts may occur in the scanned image as shown in Fig. 1(a). The artifacts are high frequency screen like patterns. They lead to low aesthetic quality and produce Moiré effect in the hard copies if the scanned images are reprinted [1]. Since halftoning is a process with information loss, the inverse process is an ill-posed problem. To reconstruct high quality contone image from halftone image, a variety of different inverse halftoning methods have been proposed.

Even though most of these methods recover contone images with sharp edges and details from binary halftone images, they cannot be used to descreen scanned halftone images [2], [3]. The scanned halftone images are gray scale, but these methods are usually designed for binary halftone images. Methods like lookup table based methods only take binary halftone images as input [4]–[6]. Some methods require information about forward halftoning process to constrain the ill-posed problem [7]–[9]. In Li’s method [9], the exact forward halftoning algorithm is needed to project the estimated contone image back to the binary halftone. However, such information usually can hardly be extracted from the scanned images currently. Although a few methods [10]–[12] are designed to learn information about halftone algorithm for halftone images, they only work on binary halftone images.

Some methods designed for binary halftone show possibilities of application to scanned halftone descreening. The wavelet based methods [13]–[15] can produce high quality contone images for binary halftone generated by error diffused method. However, the performance of these methods is highly dependent on the type of halftone patterns. As shown in [13], the performance on scanned halftone image is limited. The sparse representation based methods [16], [17] solve the inverse problem of binary halftone by training two dictionaries or the from contone samples and their corresponding binary halftones. The least-mean-square (LMS) filtering methods [18], [19] learn the weight kernel of the LMS filter from the binary halftone and contone pairs. The binary halftone image is naturally aligned with its contone version, while the scanned halftone is different with its contone in resolution and position. Therefore, to generate sample pairs for scanned image descreening, high-end printing and scanning equipments are needed to produce high quality contone images and registration must be done to align the contone image and corresponding scanned halftone.

To descreen the gray scale halftone images, the following methods were developed. Jaimes et al. [20] suppressed halftone patterns using box-filter, whose size was adaptively selected by the detected screen frequency of the scanned images. However, the box-filter decreases image sharpness without awareness of image content. Siddiqui et al. developed two different methods to descreen scanned halftone images [1], [21], i.e., the training based method and the local gradient based method. In the training based method [1], scanned halftone images and corresponding contone images were aligned to make training pairs using image registration and a super-resolution method was adopted to learn the relation between the scanned halftone and corresponding contone. Although this method preserves sharpness of edges very well, it cannot recover high quality smooth regions. And for each different printing and scanning resolution, sample image pairs must be collected for training to optimize parameters in the super-resolution algorithm. The second method estimates contone value of each pixel using local gradient information [21]. This method is highly efficient in computation but cannot
remove halftone patterns along strong edges.

Compared to binary halftone images generated by halftoning algorithms, the artifacts in scanned halftone images are more than pure halftone patterns. When the binary halftone images go through the printing process, the original halftone patterns are reproduced on paper with distortions and noise because of mechanical disturbance and other unstable factors in printers. And when such printed hard copies are scanned, the scanned halftone images usually contain distorted halftone patterns and random noise. In the existing literature, the artifacts in scanned images are called screening patterns [3] or halftone noise [13]. They mainly refer to halftone patterns in the scanned images. In conventional methods, distortions and random noise caused by printing are seldom taken into consideration. So the contone images recovered by conventional methods are usually noisy, especially in smooth regions.

Currently, there are mainly three types of halftoning algorithms: clustered dot [22], [23], dispersed dot [24], [25] and error diffusion [26], [27]. Even though cluster dot halftoning algorithms lose more information of the original contone image than the others, most EP printers still use the clustered dot halftoning technique. The reason is that EP printers cannot produce very delicate halftone patterns generated by the dispersed dot or error diffusion halftoning. In this paper, a new model of scanned images with clustered dot halftone patterns is built. This model considers the distortions of halftone patterns and random noise brought in by printing.

Based on this model, a new descreening method is proposed to obtain contone images with both clean smooth regions and sharp edges from the scanned halftone images. First, random noise in the scanned image is reduced by using the image redundancy based denoising algorithms [28], [29]. Then an adaptive filtering algorithm is developed to remove halftone patterns in the scanned image and preserves as much details as possible. A 2D anisotropic Gaussian kernel is used for the adaptive filter. The screen frequency of the scanned image is used to determine the kernel size. Local gradient information is extracted to find out the direction and salience of edges in the scanned images. The shape of the kernel is adjusted adaptively by the edge direction and salience. Finally, an edge preserving filter is adopted for edge sharpening. Fig. 1 gives an example of the proposed descreening method. As shown in this figure, the proposed method produces both sharp edges and clean smooth regions.

The rest of this paper is organized as follows. In Section II, a new model of scanned images with clustered dot halftone is proposed. In Section III, the proposed scanned image descreening method is explained in detail. Section IV gives the experimental results. Finally, Section V makes concluding remarks and points out the future work.

II. MODEL OF SCANNED IMAGES

Fig. 2 illustrates the clustered dot halftone pattern widely used by EP printers. A halftone cell is the the basic meaningful unit in the clustered dot halftone patterns. The screen angle is the angle at which halftone patterns run in relation to the horizontal. The screen frequency $f_s$ describes how many halftone cells are printed in one inch along the screen angle. Since the scanned image is comprised of a large number of halftone cells, a model of a halftone cell should be first established before modeling the scanned image.

As shown in the close-up view of Fig. 2, the binary halftone cell of clustered dot halftone consists of black pixels at the center. Although the black pixels may form different shapes in halftone cells of the binary halftone image, EP printers usually mix the black pixels together and print an ink dot for each halftone cell. The size of the ink dot is determined by the number of black pixels in the halftone cell. And the binary halftone cell degenerates into a gray scale dot due to the distribution of ink. The further the position is away from the center of halftone cell, the less ink it is covered with. We model the distribution of ink in a halftone cell with a Gaussian
function. The energy absorbed by the ink in the halftone cell is modeled as follows:

$$q(v, u, \sigma) = \lambda \exp\left(-\frac{||v - u||^2}{2\sigma^2} + \epsilon_c\right) + \epsilon_n$$  \hspace{1cm} (1)$$

where $q(v)$ represents the energy absorbed by ink at position $v$, $\lambda$ is a constant absorbing factor, $u$ is the center of halftone cell, $\sigma$ controls the size of the clustered dot in the halftone cell, which depends on the value of the corresponding contone pixel, $\epsilon_c$ and $\epsilon_n$ are random factors caused by printing and scanning, $\epsilon_c$ is the random variable describing that the halftone cell center deviates from the screen grid because of printing inaccuracy. $\epsilon_n$ represents the random noise caused by printing or paper. Both $\epsilon_c$ and $\epsilon_n$ are assumed to follow the normal distribution which is usually used to describe the distribution of random variables in real systems.

The scanned image is acquired by quantizing the energy of light reflected by the hard copy. The more energy the ink absorbs, the darker the acquired pixel gets. Denoting $I_s$ as the scanned image, the scanned image can be modeled as follows:

$$I_s(v) = 1 - \sum_k q(v, u_k, \sigma_k)$$  \hspace{1cm} (2)$$

where $k$ is the index of the halftone cell, and $u_k$ and $\sigma_k$ are respectively the center and scale factor of the $k$th halftone cell. If we substitute (1) into (2), the model of scanned image can be rewritten as:

$$I_s(v) = 1 - \lambda \sum_k \exp\left(-\frac{||v - u_k||^2}{2\sigma_k^2} + \epsilon_c\right) + \epsilon_n$$  \hspace{1cm} (3)$$

In this model, the intensity of blank paper is assumed to be 1. The clustered dot halftone patterns are represented with Gaussian components with different centers and scale factors. Denoting the coordinate of halftone cell center $u_k$ by $(x_{\mu_k}, y_{\mu_k})$, the screen grid can be described as follows:

$$x_{\mu_k} = x_{\mu_0} + mT \cos \gamma + nT \sin \gamma$$  \hspace{1cm} (4)$$

$$y_{\mu_k} = y_{\mu_0} + mT \sin \gamma + nT \cos \gamma$$  \hspace{1cm} (5)$$

$$T = f_s^{-1}$$  \hspace{1cm} (6)$$

where $f_s$ is the screen frequency of halftone patterns, $T$ is the halftone cell size, $\mu_0$ is the center of the halftone cell locating at the origin of the screen grid, $(m, n)$ is the two dimensional index of the $k$th halftone cell and $\gamma$ is the screen angle.

![Diagram](https://via.placeholder.com/150)

**Fig. 3.** Overview of the proposed descreening method.

### III. SCANNED IMAGE DESCREENING

According to the proposed model, two tasks must be accomplished when recovering a contone image from the scanned image: random noise reduction and halftone pattern removal. Fig. 3 is an overview of the proposed descreening method. Firstly, the image redundancy based denoising algorithm is adopted to reduce random noise and attenuate printing distortions. Secondly, features (including screen frequency and local gradient) are extracted from the denoised image to provide necessary information for adaptive filtering. Then, adaptive kernels are generated based on these features and used to filter out the halftone patterns in the denoised image. Finally, an edge preserving filter is used to enhance the sharpness of edges to produce the final output contone image.

#### A. Image Redundancy Based Denoising

Recently, image redundancy based denoising algorithms, such as non-local mean filtering (NLM) [28] and block matching and 3D transform domain collaborative filtering (BM3D) [29], are very popular and effective to reduce Gaussian white noise in natural images. These image redundancy based denoising algorithms consist of four basic steps: 1) partition the input noisy image into overlapped patches, 2) search for similar patches, 3) estimate the clean patch based on these similar patches, and 4) aggregate the clean patches to generate the output noise-free image. In this paper, the image redundancy based denoising method is adopted to reduce random noise in the scanned image and attenuate printing distortions of the halftone patterns. Our experiments show that the NLM algorithm and the BM3D algorithm can produce equivalent denoised results, but the BM3D algorithm is chosen for the sake of computational efficiency. The BM3D algorithm takes advantage of image redundancy, i.e., the repeated image content, to reduce random noise. In the scanned image, the most repeated image content is the periodic halftone patterns. By searching for similar image patches, the BM3D algorithm can find patches with similar halftone patterns. When these patches are transformed into the 3D transform domain, the aligned halftone patterns produce responses strong enough to survive in the process of collaborative filtering. On the contrary, the random noise in these patches are reduced because their weak responses in the 3D transform domain are suppressed.
Apart from the random noise, the fluctuation of halftone cell centers can also be attenuated in this process, because the 3D collaborative filtering only preserves correlated image components and attenuates those that can not be aligned. So it is very helpful to unify the halftone cells in the smooth region. Therefore, the BM3D algorithm can reduce the random noise in the scanned image as well as attenuate printing distortions to the halftone patterns. For more details of the BM3D algorithm, please refer to [29].

B. Feature Extraction

1) Screen Frequency Estimation: The screen frequency of the scanned image is decided by both the LPI (Lines Per Inch) of the printer and the DPI (Dots Per Inch) of the scanner involved in the printing and scanning process. If the LPI and DPI are known priori, the screen frequency of the scanned image can be calculated as follows:

\[
f_s = \frac{L_p \cos \frac{\pi}{4}}{R_s}
\]

(7)

where \(L_p\) is the LPI parameter of the printer, \(R_s\) is the DPI parameter of the scanner, and \(\pi/4\) is the commonly used screen angle for gray-scale printing. Since the most important application of the descreening is the print-to-scan path in the multi-function printer, the above parameters of printing and scanning are assumed to be already known.

2) Local Gradient Extraction: Apart from the screen frequency, local gradient information of image content is also used to tune the parameters of the adaptive filter. Since the scanned image is highly cluttered with halftone patterns in the background, local gradient information can not be accurately extracted with conventional methods. To estimate the edge direction and salience, a more robust method called steerable filter [30] is adopted. The kernel of the steerable filter along direction \(\varphi\) can be described as follows:

\[
G^0(x, y) = \frac{\partial}{\partial x} \exp \left( \frac{x^2 + y^2}{2\sigma_G^2} \right)
\]

(8)

\[
G^{\pi/2}(x, y) = \frac{\partial}{\partial y} \exp \left( \frac{x^2 + y^2}{2\sigma_G^2} \right)
\]

(9)

\[
G^\varphi(x, y) = \cos \varphi G^0(x, y) + \sin \varphi G^{\pi/2}(x, y)
\]

(10)

where \(x, y = -r, -r+1, ..., 0, ..., r\) and \(\sigma_G\) is the scale factor. Note that the kernel along an arbitrary direction is a linear combination of two basic kernels \(G^0\) and \(G^{\pi/2}\). The radius \(r\) and scale factor \(\sigma_G\) of the steerable filter are determined by the screen frequency of scanned image. They are calculated as follows:

\[
r = \lambda_r f_s
\]

(11)

\[
\sigma_G = \lambda_s r
\]

(12)

where \(\lambda_r\) is the kernel radius factor and \(\lambda_s\) is the kernel scale factor. The two factors are selected empirically in the experiments.

The gradient magnitude along direction \(\varphi\) for pixel \(u(s_1, s_2)\) in the denoised image can be calculated as follows:

\[
g_\varphi(s_1, s_2) = \left| \sum_{x=-r}^{r} \sum_{y=-r}^{r} G^\varphi(x, y) u(s_1 + x, s_2 + y) \right|
\]

(13)

For each pixel in the denoised scanned image, gradient magnitudes along different directions are calculated by using the steerable filters. In the proposed method, gradient magnitudes along several different directions are calculated independently. As shown in Fig. 4, steerable filters along \(-\pi/3, -\pi/4, -\pi/6, 0, \pi/6, \pi/4, \pi/3, \pi/2\) are used to extract gradient magnitudes along these 8 directions.

C. Adaptive Filtering

In the proposed descreening method, an adaptive filtering strategy is developed to predict the contone value of the scanned image pixel by pixel. The kernel of the adaptive filter is an anisotropic 2D Gaussian function. With the features extracted from the scanned image, the parameters of the kernel function are tuned automatically. The kernel size is adaptively adjusted for scanned images at different resolution, and the kernel can change its shape to adapt to different image content. When the target pixel is in the smooth region, its kernel expands to the same scale along all directions; when the target pixel is on an edge, the kernel expands more along the edge than other directions.

The adaptive filter can be written as follows:

\[
v(s_1, s_2) = \sum_{x=-r}^{r} \sum_{y=-r}^{r} \omega(x, y) u(x + s_1, y + s_2)
\]

(14)

where \(r\) is the radius of local window centered at target pixel \(u(s_1, s_2)\), and \(\omega\) is the kernel of the adaptive filter, i.e., the weight for each pixel in the local window. The kernel is a 2D anisotropic Gaussian function:

\[
\omega(x, y) = \frac{1}{z} \exp(-ax^2 - bxy - cy^2)
\]

(15)

The coefficients \(a, b\) and \(c\) in (15) are defined as follows:

\[
a = \frac{\cos^2 \theta}{2\sigma_1^2} + \frac{\sin^2 \theta}{2\sigma_2^2}
\]

(16)

\[
b = \frac{\sin \theta \cos \theta}{2\sigma_1^2} - \frac{\sin \theta \cos \theta}{2\sigma_2^2}
\]

(17)

\[
c = \frac{\sin^2 \theta}{2\sigma_1^2} + \frac{\cos^2 \theta}{2\sigma_2^2}
\]

(18)
where $\theta$ is the angle the edge runs in relation to the horizontal detection, $\sigma_1$ is the scale factor along edge direction and $\sigma_2$ is the other scale factor along the orthogonal direction.

The adaptiveness of the proposed method is achieved by automatic tuning of the aforementioned four parameters, i.e., neighborhood radius $r$, edge direction $\theta$, scale factors $\sigma_1$ and $\sigma_2$. The neighborhood radius $r$ is decided by the screening frequency of the scanned image; the edge direction $\theta$ is selected as the direction with the minimum gradient magnitude; the scale factor $\sigma_1$ and $\sigma_2$ are decided by the detected screen frequency and gradient magnitudes along different directions. The neighborhood radius $r$ of the adaptive filter kernel is the same as the kernel radius of the steerable filter calculated with (11); and the scale factor $\sigma_1$ is the same with $\sigma_G$ calculated with (12). Once $\lambda_1$ and $\lambda_2$ are fixed, the parameters $r$ and $\sigma_1$ are only related to the screen frequency $f_s$. And the screen frequency $f_s$ is a global feature of the scanned image. Therefore, the parameters $r$ and $\sigma_1$ keep the same value for all pixels of the scanned image. But for scanned images with different screen frequencies, the values of $r$ and $\sigma_1$ may change.

With gradient magnitude along different directions, the edge direction is assumed to be the direction with minimum gradient magnitude. The edge direction $\theta$ and the other scale factor $\sigma_2$ of the adaptive kernel are dependent on the local gradient information of the image content and vary from pixel to pixel in the same scanned image. They are calculated as follows:

$$
\theta(s_1, s_2) = \arg \min_{\varphi} g_{G}(s_1, s_2) 
$$

$$
\sigma_2(s_1, s_2) = \sigma_1 \left/ \left(1 + \frac{g_{med}}{g_{max} - g_{min}} \frac{g_{max} g_{min} + g_{med}}{\eta + g_{med} g_{min} + g_{med}} \right) \right.
$$

where $\eta$ is an edge sensitivity factor, $g_{max}$, $g_{min}$ and $g_{med}$ are the maximum, minimum and median of gradient magnitude along different directions at pixel $(s_1, s_2)$ respectively. Considering the extreme values of $\eta$, we obtain the following limitations from (20).

$$
\lim_{\eta \to 0} \sigma_2(s_1, s_2) = \sigma_2 \frac{g_{min} + g_{med}}{g_{max} + g_{med}}
$$

$$
\lim_{\eta \to \infty} \sigma_2(s_1, s_2) = \sigma_1
$$

As expressed in (21) and (22), the filter is anisotropic and very sensitive to edges when $\eta = 0$, whereas it degenerates into an isotropic Gaussian filter with two equal scale factors when $\eta \to \infty$. So the adaptive filter is more sensitive to edges when $\eta$ is smaller. In our implementation, the factor between $\sigma_1$ and $\sigma_2$ in (20) is quantized into 20 discrete values within $[1, 3.5]$ to improve computational efficiency.

Fig. 5 illustrates kernels used in the adaptive filter for different image content. Target pixels are marked with red dots in the denoised scanned image and the corresponding adaptive kernels are shown in close-up view. The target pixel $a$ is in a smooth region and its adaptive kernel degenerates into an isotropic Gaussian kernel. The other three target pixels $b$, $c$ and $d$ are on edges with different directions. From the kernels of target pixels $b$, $c$ and $d$, we can see that the kernel of the adaptive filter follows the direction of edges quite well.

**D. Edge-preserving Filtering**

Once the basic estimate of the contone image is obtained, the edge-preserving filtering can be performed to produce sharper edges [1]. The edge-preserving filter takes two input images: the denoised scanned image and the basic contone estimate. It first computes the kernel for each pixel in the basic estimate, and smooth the denoised scanned image with these kernels to predict contone values for the corresponding pixels in the denoised scanned image.

Theoretically, a variety of edge-preserving filters can be used in this step. Representatives include the guided filter [31], the bilateral filter [32] and the NLM filter [28]. The guided filter takes advantage of local variance, while the bilateral filter utilizes both spatial and intensity distances. The non-local means filter considers the structural similarity, a higher level image feature. Fig. 6 shows results of the proposed method using different edge-preserving filters. Compared with the descreened images with edge-preserving filtering, the edges in the descreened image without edge-preserving filtering tend to be less sharp as shown in Fig. 6(a). In terms of edge and...
detail preservation, the NLM filter is better than the other two methods as shown in the close-up views. However, the NLM method requires intensive computation to measure the similarity between local patches. Since edge produced by bilateral filter is a little sharper than the guided filter, it is adopted in the proposed descreening method.

The bilateral filter for edge preservation computes the final contone image value of pixel at \((s_1, s_2)\) as follows:

\[
h(s_1, s_2) = \frac{1}{\sigma_s^2 + \sigma_b^2} \sum_{x,y} \omega_s(x,y) \omega_b(x,y) u(s_1 + x, s_2 + y)
\]

where \(h\) is the final contone image, \(u\) is the denoised scanned image, \(z\) is a normalization factor, \(\omega_s\) and \(\omega_b\) are the spatial and brightness weights of the bilateral kernel respectively. The three components of bilateral kernel, normalization factor \(z\), space weight \(\omega_s\) and brightness weight \(\omega_b\) are calculated with (24), (25) and (26) respectively.

\[
z = \sum_{x,y} \omega_s(x,y) \omega_b(x,y)
\]

\[
\omega_s(x,y) = \exp\left(-\frac{x^2 + y^2}{2\sigma_s^2}\right)
\]

\[
\omega_b(x,y) = \exp\left(-\frac{(v(s_1 + x, s_2 + y) - v(s_1, s_2))^2}{2\sigma_b^2}\right)
\]

where \(v\) is the basic estimate of corresponding contone value, \(\sigma_s\) is the space scale factor, and \(\sigma_b\) is the brightness scale factor. The radius of the bilateral filter is the same as radius \(r\) calculated with (11) and its space scale factor \(\sigma_s\) is equal to scale factor \(\sigma_G\) calculated with (12).

IV. EXPERIMENTAL RESULTS

The proposed method is designed to descreen image components in the scanned document images. In order to evaluate performance of the proposed scanned image descreening method for complex image content, we print and scan the 63 images in the Berkeley Segmentation Database [33]. Thumbnails of some test images are shown in Fig. 7. Each original contone image is printed by printer RICOH Aficio MP4500 at 300 and 600 dpi respectively, and every hard copy is scanned by scanner Fujitsu fi-6130 at 200, 300, 400 and 600 dpi respectively.

A. Comparison with Other Methods

The performance of the proposed method is compared with four existing methods: 1) hardware friendly descreening algorithm (HFD) [21]; 2) algorithm I of training-based descreening using Gaussian filter for prediction (TBD-I) [1]; 3) algorithm II of training-based descreening using resolution synthesis for prediction (TBD-II) [1]; and 4) Jaimes et al.’s method [20]. Software of the training based method is available on the web\(^1\). We implemented the HFD method in Matlab.

All the parameters of TBD-I, TBD-II and HFD used for the experiments on image scanned at 300 dpi are listed in Table I. The values of these parameters are all set as reported in the relevant papers. The parameters of the proposed method are listed in Table II.

Results of different descreening methods on images printed at 300 dpi and scanned at 300 dpi are shown in Fig. 8. Fig. 8(a) shows the original scanned images. Fig. 8(b) shows the results obtained by the HFD method. It can be seen that the HFD method removes halftone patterns in smooth regions very well, but cannot remove halftone patterns on the edges. From the results of the TBD-I method shown in Fig. 8(c), it can be seen that the halftone patterns are removed effectively, but the edges are blurred. This is caused by the Gaussian filter used in this method for basic prediction of the contone value. The Gaussian filter is not aware of image content, so it smooths the edges and the smooth regions in the same way when it filters the halftone patterns out. The random noise in the descreened image is still visible in the smooth regions. The TBD-II method produces very sharp edges but can not remove the random noise in the smooth regions as shown in Fig. 8(d).

Results of the proposed decreening method are shown in Fig. 8(e). It can be found that the proposed method removes halftone patterns along edges as well as in smooth regions.

\(^1https://engineering.purdue.edu/~bouman/software/RSD/\)
Moreover, the edges in the results of the proposed method are sharper and clearer than the results of the TBD-I method. The reason is that the proposed adaptive filter can make its kernel fit the edges by using local gradient information. The proposed method produces edges as sharp as the TBD-II method. The TBD-II method requires training for input images scanned at different resolution, whereas the proposed method does not need any extra images for training.

Comparing the smooth regions of all the results shown in Fig. 8, we can see that the proposed method produces smooth regions with the highest perceptual quality. The proposed method matches halftone patterns in local patches and reduces random noise using the redundant information contained in these patches. However, the random noise and distortions caused by printing are not considered in the other methods. Therefore, they produce noisy smooth regions.

Since the scanned image and the original contone image are generally different in resolution and orientation due to printing and scanning, the full reference image quality assessment metrics cannot be used. Therefore, we have to turn to blind (or non-reference) image quality assessment metrics.

The Blind/Referenceless Image Spatial Quality Evaluation (BRISQUE) [34] metric is designed to assess images with either noise or blurring, the major distortions in the descreened images. Therefore, it is adopted to perform an objective evaluation of the TBD-I, TBD-II and proposed methods. Table III compares the BRISQUE scores obtained by the proposed method with the TBD-I and TBD-II methods on all the 63 images.
images with different resolutions. From Table III, it can be found that the proposed method obtains better BRISQUE scores than the TBD-I and TBD-II methods on most images.

Fig. 9 shows the results of different descreening methods on
Fig. 11. Results of the proposed method on scanned images with different printing and scanning settings. The first and third rows are scanned images, while the second and fourth rows are corresponding descreening results. (a) Printed at 300 dpi and scanned at 200 dpi. (b) Printed at 300 dpi and scanned at 300 dpi. (c) Printed at 300 dpi and scanned at 400 dpi. (d) Printed at 300 dpi and scanned at 600 dpi. (e) Printed at 600 dpi and scanned at 200 dpi. (f) Printed at 600 dpi and scanned at 300 dpi. (g) Printed at 600 dpi and scanned at 400 dpi. (h) Printed at 600 dpi and scanned at 600 dpi.

images printed at 600 dpi and scanned at 600 dpi. From Fig. 9(b), it can be found that the HFD method cannot remove halftone screens in scanned images with very large halftone cells. The reason may be the HFD method performs weighted average filtering in a $7 \times 7$ local window. To descreen scanned images with different halftone cell sizes, the HFD method may need to design new masks with different sizes. In Fig. 9(c), the TBD-I method removes halftone screen clearly but the edges are not sharp. In Fig. 9(d), slight screen can be perceived in the results of the TBD-II method but the edges are sharper than the TBD-I method. In Fig. 9(e) slight screen also occurs, but the proposed descreening method still produces the best perceptual quality due to unified smooth regions and sharp edges.

The proposed method is also compared with Jaimes et al.’s method [20]. The original scanned image and result of Jaimes et al.’s method are obtained from their paper [20]. Figs. 10(a) and (c) show results of Jaimes et al.’s method and the proposed method respectively. Fig. 10(b) compares the results of the two methods in close-up views. From the comparison, we can see that the proposed method produces both cleaner smooth regions and sharper edges.

It should be pointed out that in this case the screen frequency of the scanned image is estimated by locating the halftone response peak in the Discrete Cosine Transform spectrum of the scanned image. Generally, if the printing and scanning priori of the scanned image are not sufficient, other effective algorithms like [20] can also be integrated into the framework of the proposed descreening method to estimate the screen frequency.

B. Different Printing and Scanning Settings

Fig. 11 demonstrates results of the proposed method on real scanned images with different printing and scanning settings. Figs. 11(a), (b), (c) and (d) show results of the proposed method on images printed at 300 dpi and scanned at 200, 300, 400 and 600 dpi respectively. Figs. 11(e), (f), (g) and (h) are results of the proposed method on images printed at 600 dpi and scanned at 200, 300, 400 and 600 dpi respectively. It should be emphasized that all these results are produced using the same parameters shown in Table II.

By comparing the results for scanned images scanned at 200, 300, 400 and 600 dpi, it can be seen that the edge sharpness of our result decreases a little as the scanning resolution increases. As the scanning resolution increases, the halftone patterns are more and more salient. Then adaptive filters with a larger kernel are needed to suppress the halftone patterns. By comparing the results on images printed at 300 dpi and 600 dpi, it can be found that the contrast of results for the scanned images printed at 600 dpi is lower than that printed at 300 dpi. This is caused by the printer the contrast of original scanned images also decreases when print resolution increases. In Fig. 11(e), vertical stripes appear in the scanned image. Some stripes are filtered out by the proposed method while some are still perceptible but much slighter than the original.

Generally speaking, the proposed method works very well on the test images scanned at 300 dpi, and produces promising
results for the others without tuning parameters manually.

C. Effect of Parameters

The above experimental results show that the proposed algorithm works very well using the parameters in Table II. In this section, experiments are conducted to study the effect of different parameter values. In each experiment, only one parameter varies while others keep the same value as shown in Table II.

As the BM3D denoising algorithm [29] is adopted to remove printing noise, the estimated noise level is the first parameter. Fig. 12 shows our results using different values of estimated noise level. It can be seen that when the value of noise level increases, more and more noise is removed. But on the other hand, less spatial details are preserved. To strike a balance, we select the noise level parameter as 15 empirically.

Fig. 13 shows the results of the proposed method with different values of the kernel radius factor $\lambda_r$. From Fig. 13(a), it can be seen that the halftone patterns cannot be removed if the kernel size of the adaptive filter is too small.
When the kernel radius factor is 0.85, halftone patterns are hardly perceptible in both smooth regions and detail regions in recovered contone image. As this factor goes on increasing, more and more details are filtered out. So the kernel radius factor λ is selected as 0.85 to preserve as much details as possible.

Fig. 14 shows results of the proposed algorithm with different values of the kernel scale factor λs. This parameter is set to 0.5 as a trade-off between halftone pattern removal and detail preservation. Fig. 15(a) shows histograms of the values of σ1/σ2 with different η values. It can be seen that more and more values of σ1/σ2 approach 1 when η increases, which means more and more adaptive filters becomes isotropic as described with (22). In Fig. 15(b), some faint screen artifacts still occur in the smooth regions near edges when η is too small. As shown in Figs. 15(c), the artifacts disappear with the increasing of η. This is because more and more adaptive kernels degenerate into isotropic filters as shown in Fig. 15(a). As can be derived from (20), the degeneration of adaptive filters happens to the pixels in smooth regions first. No degeneration will happen to pixels with very large gradient magnitude until η is extremely large.

The brightness scale factor σb of bilateral filter controls intensity smoothness of the recovered contone image. Fig. 16 demonstrates the effect of this parameter on results of the proposed method. As σb increases, more and more trivial screen artifacts are smoothed. It is a good choice to set σb to the value around 20 on account of detail preservation.

![Image](https://via.placeholder.com/150)

**Fig. 16.** Results of the proposed method using different values of brightness scale factor σb. The brightness scale factors σb of (a)-(d) are 5, 15, 20 and 30 respectively.

D. Computational Efficiency

Table IV shows the time used by different methods to descreen a gray scale scanned image with 830 × 540 pixels on PC with 2.2GHz CPU and 2G RAM. To avoid loops in Matlab, the adaptive filtering and bilateral filter is implemented in C++. Time consumption on each step of the proposed method is as follows: the BM3D denoising costs 11s, feature extraction 0.7s, adaptive filtering 0.1s and bilateral filtering 2s. The high computational efficiency of the adaptive filter is achieved by approximating the adaptive filter with an adaptive kernel family, which is comprised of 160 kernels with different θ and σ1 values. For real-time applications, there are two ways to reduce computation of the proposed method.

One way is to substitute the BM3D with a simple denoising algorithm. However, it should be emphasized that the new denoising algorithm shall finish two tasks as we proposed in Section III-A, i.e., attenuating distortions to halftone cells and reducing random noise. Denoising algorithms working on pixel level are not suitable for these tasks, because they usually destroy the screen structure of the halftone image while reducing random noise.

The other way is to eliminate random noise reduction block altogether. Fig. 17(d) shows the result of the proposed method with the BM3D turned off. Although the random noise will reduce the perceptual quality of the descreened image, the proposed method without the BM3D still produces very sharp edges. The fast version of the proposed method produces similar results with the TBD-II method as shown in Fig. 17(b).

**V. CONCLUSIONS AND FUTURE WORK**

In this paper, we build a model of scanned image with clustered dot halftone patterns, which considers distortions and noise caused by printing. The proposed adaptive filtering based descreening method can produce high quality continuous tone images for scanned halftone images. Results of the proposed method on real scanned images demonstrate very sharp edges and clean smooth regions. The proposed method detects screen frequency of the scanned image and takes advantage of the local gradient information to perform adaptive filtering. It can descreen scanned images at different printing and scanning resolutions without tuning parameters manually.

The proposed method can be not used to descreen color scanned images directly, because the color scanned image is composed of four tangled screens with different angles in CMYK color space. How to untangle the four screens and descreen each one independently will be one of our tasks in the future. Other future tasks may include design of more robust screen frequency detector and extraction of more features to instruct adaptive filtering.

**ACKNOWLEDGMENT**

The authors are very grateful to Dr. Brendt Wohlberg for the useful discussions and constructive suggestions. The authors also thank the anonymous reviewers for their valuable comments to improve the paper.
REFERENCES


Bin Sun received the B.Sc. degree from Hunan University, Changsha, China, in 2010. He is currently working towards the Ph.D. degree at the Hunan University. His research interests are image super-resolution and document image analysis.

Shutao Li (M’07) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Hunan University, Changsha, China, in 1995, 1997, and 2001, respectively. In 2001, he joined the College of Electrical and Information Engineering, Hunan University. From May 2001 to October 2001, he was a Research Associate with the Department of Computer Science, Hong Kong University of Science and Technology, Kowloon, Hong Kong. From November 2002 to November 2003, he was a Postdoctoral Fellow with the Royal Holloway College, University of London, Egham, U.K., working with Prof. J.-S. Taylor. From April 2005 to June 2005, he was a Visiting Professor with the Department of Computer Science, Hong Kong University of Science and Technology. He is currently a Full Professor with the College of Electrical and Information Engineering, Hunan University. He has authored or coauthored more than 150 refereed papers. His professional interests are information fusion, image processing, and pattern recognition.

Jun Sun received the B.Sc. degree in electrical engineering from Hunan University, Changsha, China, in 1995, and the M.Sc., and Ph.D. degrees in pattern recognition and machine intelligence from Tsinghua University, Beijing, China, in 1998, and 2001, respectively. In 2002, he joined Fujitsu Research and Development Center (FRDC). He is currently the deputy director of information technology laboratory in FRDC. He has authored or coauthored more than 50 papers. His professional interests are pattern recognition and document image processing.