PAPER

Clipping-Free Halftoning and Multitoning Using the Direct Binary Search

Xia ZHUGE†, *Student Member and* Koji NAKANO†, *Member*

SUMMARY Halftoning is an important process to convert a gray scale image into a binary image with black and white pixels. The Direct Binary Search (DBS) is one of the well-known halftoning methods that can generate high quality binary images for middle tone of original gray scale images. However, binary images generated by the DBS have clippings, that is, have no tone in highlights and shadows of original gray scale images. The first contribution of this paper is to show the reason why the DBS generates binary images with clippings, to clarify the range of tone in original images that may have clipping, and to present a clipping-free DBS-based halftoning algorithm. The key idea is to apply the ordered dither using a threshold array generated by DBS-based method, to highlights and shadows, and then use the DBS. The second contribution is to extend the DBS to generate L-level multitone images with each pixel taking one of the intensity levels $\frac{0}{L-1}$, $\frac{1}{L-1}$, ..., $\frac{L-1}{L-1}$. However, clippings appear in highlights, middle tone, and shadows of generated L-level multitone images. The third contribution of this paper is to modify the multitone version of the DBS to generate a clipping-free L-level multitone images. The resulting multitone images are so good that they reproduce the tones and the details of the original gray scale images very well.

key words: *Image processing, Halftoning, Multilevel halftoning, Direct binary search*

1. Introduction

A gray scale image is a two dimensional matrix of pixels taking a real number in the range $[0, 1]$. Usually a gray scale image has 8-bit depth, that is, each pixel takes one of the real numbers $\frac{0}{255}, \frac{1}{255}, \ldots, \frac{255}{255}$, which correspond to pixel intensities. *A binary image* is also a two dimensional matrix of pixels taking a binary value 0 (black) or 1(white). *Halftoning* is an important process to convert a gray scale image into a binary image [2], [11], [12]. This process is necessary when a monochrome or color image is printed by a printer with limited number of ink colors.

A multitone image is an intermediate of a gray scale image and a binary image [14], [16]. In an L-level multitone image, each pixel takes one of the intensity levels $\frac{0}{L-1}$, $\frac{1}{L-1}$, ..., $\frac{L-1}{L-1}$. Usually, L takes small integers such that $L = 3, 4$, or 5. The process of *multitoning* is to generate a multitone image for a given gray scale image.

Manuscript received January 15, 1999.
Manuscript revised November 27, 2002.
Final manuscript received October 29, 2003.

†The authors are with the Department of Information Engineering, Hiroshima University

Multitone images are used to print images using a inkjet printer with light color inks to get photo quality printing results [6]. For example, some inkjet printers support light black ink in addition to black ink. In the printing workflow, an original gray scale image is converted to a 3-level multitone image. For pixels with intensity $\frac{2}{2}$, black ink is injected. If the pixel intensity is $\frac{1}{2}$, light black ink is used. The printing results can be improved, because light black ink reproduces middle tone better than black ink.

Further, some inkjet printers [13] support multi-size dots, which can be achieved by adjusting the amount of ink injected from the nozzle. Suppose that each nozzle can inject $1pl$ (pico liter), $2pl$, and $3pl$ ink. To print an original gray scale image using such a printer, it is converted to a 4-level multitone image. For pixels with intensity $\frac{1}{3}$, $\frac{2}{3}$, and $\frac{3}{3}$, the nozzle injects $1pl$, $2pl$ and $3pl$ ink, respectively. Since the size of dots reproduces the original tone, the resulting image using multi-size dot printers is better than that of single-size dot printers.

Many halftoning techniques including Error Diffusion [7], Dot Diffusion[8], Ordered Dither using the Bayer threshold array[5] and the Void-and-Cluster threshold array [15], Direct Binary Search (DBS) [1], [3], Local Exhaustive Search (LES)[9], [10], have been presented. The most well-known halftoning algorithm is the Error Diffusion [7] method that propagates rounding errors to unprocessed neighboring pixels according to some fixed ratios. The Error Diffusion preserves the average intensity level between the original input image and the binary output image. Since the Error Diffusion produces good results, many inkjet printers are using this technique [13]. However, the Error Diffusion may generate worm artifacts, sequences of pixels like a worm, especially in the image areas of flat intensity. The readers should refer to Figure 3 for resulting images for a ramp gray scale image that are generated by various halftoning/multitoning method including the Error Diffusion.

The Ordered Dither [5] uses a threshold array to generate a binary image from an original gray scale image. Each pixel of the original gray scale image is compared with an element of the threshold array. From the result of the comparison, the pixel value of the corresponding pixel of the binary image is determined. Binary images generated by the Ordered Dither method using the Bayer threshold array [5] have artifacts with periodic dots arranged in a two dimensional grid as shown in Figure 3.

It is known that, in many cases, the DBS [1], [3] generates better quality images than the Error Diffusion. The key idea of the DBS is to find a binary image whose projected image onto human eyes is very close to the original image. The projected image is computed by applying a Gaussian filter, which approximates the characteristic of the human visual system. Let the total error of the binary image be the sum of the differences of the intensity levels over all pixels between the original image and the projected image. In the DBS, a pixel value is toggled if the resulting image has smaller total error. Also, neighboring pixel values are swapped if the total error of the resulting image decreases. The DBS generates a sharp binary image, especially, for middle tone areas. However, the generated binary image by the DBS has no tone in highlights and shadows. Figures 3 and 4 show the binary images generated by the DBS. The resulting image has *clippings*, that is, highlights and shadows have no minority pixels and lose the tone of the original image. For example, several columns from the leftmost of Figure 3 have no white pixel, although the original image has tone. Also, there is no black pixel in several columns from the rightmost and a pseudo border line appears.

Halftoning methods are required to generate binary images reproducing the tone of original gray scale images. Thus, *the linearity of the intensity level* of binary images that we define next is important. Suppose that the average intensity level of a small region of s pixels in an original gray scale image is p $(0 \leq p \leq 1)$. The corresponding region of the generated binary image should have nearly ps white pixels out of s pixels to reproduce the intensity level. If this is the case, the linearity of the intensity level is preserved. More generally, the linearity can be confirmed using flat tone images as follows: Let A_p $(0 \leq p \leq 1)$ denote a flat tone gray scale image of size $n \times n$ with every pixel taking intensity level p. Suppose that a binary image generated for A_p using some halftoning method. Clearly, if the binary image has s white pixels, its average intensity level is $\frac{s}{n^2}$. Let $f(p)$ $(0 \leq p \leq 1)$ denote the average intensity level of the binary image obtained from A_p. The halftoning method preserves *the linearity of the intensity level* if $f(p) \approx p$, that is, the intensity level is correctly reproduced for all intensity levels. Clearly, Error Diffusion satisfies the linearity, because error of the intensity level is compensated by diffusing it. However, DBS generates binary images with clipping, that is $f(p) = 0$ for small p and $f(p) = 1$ for large p. Hence, DBS does not preserve the linearity and cannot reproduce the tone of highlights and shadows. Figure 2 shows the graphs of f for DBS and our DBS-based halftoning method. From the figure, we can see that our DBS-based halftoning method preserves the linearity while DBS does not preserve it.

For most printing devices, black pixels gain by dot-gain [10]. In other words, the average intensity level of the actual printed image is smaller than that of a binary image used for printing. If this is the case, the intensity levels of an original gray scale image are calibrated such that the actual printed image reproduces the intensity of the original gray scale image correctly. For example, suppose that intensity level 254/255 of an original gray scale image is adjusted to 1023/1024. After that, the adjusted gray scale image is converted to the binary image. Clearly, the binary image thus obtained have fewer black pixels and the average intensity is 1023/1024. However, black pixels gain by dot-gain, and the intensity level of the actual printed image will increase to 254/255. This means that, halftoning methods are required to generate the binary image with average intensity level 1023/1024. If this is not possible, actual printed images cannot reproduce intensity level 254/255, and should have tone jumps.

The first contribution of this paper is to clarify the reason why the DBS generates binary images with clippings and present a new DBS-based halftoning method that generates clipping-free binary images. The key idea is to apply the Ordered Dither method using a threshold array generated by the DBS to highlights and shadows of an original gray scale image. We preserve minority pixels, that is, black pixels in the highlights and white pixels in the shadow areas, and apply DBS to the whole image. The resulting binary images have no clipping and reproduce the original tones very well. Further, our DBS-based halftoning preserves the linearity of intensity levels. In general, visually pleasing halftone textures are perceived as smooth, contain a large variety of patterns, and exhibit accurate tone rendition [20]. In other words, our resulting binary images also have high texture quality.

There are several methods proposed to deal with the clipping problem in other papers. In paper [17], Pulse Density Modulation (PDM) and DBS are combined to generate a binary image. Halftone areas of the image are halftoned using DBS, and PDM is used in highlight and shadow areas. Since the ordered dither used in our halftoning method can generate more uniformly distributed dot pattern, our halftoning method is a better choice to be used in the highlight and shadow areas. In paper [18], tone-dependent Human Visual System (HVS) models are used in different tone areas to get binary images with few clippings. In paper [19], a pair of two-component Gaussian models with two sets of fixed parameters are combined through a pair of tone-dependent weights to generate binary images with few clippings. However, since this method uses a Gaussian-based filter, all the clippings cannot be removed and the linearity is not preserved. In particular, it is impossible to generate a binary image that reproduces the intensity level 1023/1024, which may be required by printers with gaining dots.

Our second contribution is to extend the DBS to generate an L-level multitone image. The key idea to generate a multitone image is as follows. Let p be the intensity of a pixel of the original gray scale image, and i be an integer such that $\frac{i}{L-1} \leq p \leq \frac{i+1}{L-1}$ holds. The intensity of the corresponding pixel of the binary image is rounded to $\frac{i}{L-1}$ or $\frac{i+1}{L-1}$. We use the DBS to determine if each pixel is rounded to $\frac{i}{L-1}$ or $\frac{i+1}{L-1}$. Figure 3 and 5 show the resulting 3-level multitone images. The middle tone areas of pixels whose intensities are close to $\frac{1}{2}$ have no tone. In general, for L-level multitoning, the resulting multitone image has no tone in the pixel areas whose intensities are close to $\frac{i}{L-1}$ for every integer i.

The third contribution of this paper is to present a clipping-free DBS-based multitoning method. To reproduce the tone correctly, we use the Ordered Dither for the range of intensity that may have clippings, and then apply the DBS. The threshold array used by the Ordered Dither is generated by a DBS-based technique. Using our multitoning method, we can generate a clipping-free high quality multitone image that reproduces the tones and the details of the original gray scale image.

This paper is organized as follows. Section 2 reviews the Ordered Dither method and the Direct Binary Search method to obtain binary images. In Section 3, we show the reason why the DBS generates a binary image with clippings, and clarify the range of intensity that may have no tone. In Section 4, we present our clipping-free DBS-based halftoning method. Section 5 presents two multitoning methods using the Ordered Dither and the DBS to generate L-level multitone images. Section 6 presents our DBS-based multitoning method that generates clipping-free L-level multitone images. Section 7 offers conclusion.

2. The Ordered Dither and The Direct Binary Search

The main purpose of this section is to review the Ordered Dither [5], [15] and the Direct Binary Search [1], [3], which are key ingredients of our new DBS-based halftoning/multitoning methods.

Suppose that an original gray scale image $A = (a_{i,j})$ of size $n \times n$ is given[†], where $a_{i,j}$ denotes the intensity level at position (i,j) $(0 \leq i,j \leq n-1)$ taking a real number in the range $[0,1]$. The goal of halftoning is to find a binary image $B = (b_{i,j})$ of the same size that reproduces the original image A, where each $b_{i,j}$ is either 0(black) or 1(white). The Ordered Dither uses a threshold array $T = (t_{i,j})$ of size $m \times m$, with each element taking a real number $\frac{0}{255}, \frac{1}{255}, \ldots,$or $\frac{254}{255}$. More specifically, the pixel value of each pixel $b_{i,j}$ is determined by the following formula:

$$b_{i,j} = \begin{cases} 0 & \text{if } a_{i,j} \leq t_{i \bmod m, j \bmod m} \\ 1 & \text{if } a_{i,j} > t_{i \bmod m, j \bmod m} \end{cases}$$

Note that the threshold value never takes $\frac{255}{255}$, because $b_{i,j}$ is always 0 if $t_{i \bmod m, j \bmod m} = \frac{255}{255}$. The Bayer halftoning uses the Bayer threshold array which is defined recursively [5]. As shown in Figure 3, the resulting binary image generated by the Ordered Dither using the Bayer threshold array has artifacts with periodic dots.

The idea of the DBS is to measure the goodness of the output binary image B using the Gaussian filter that approximates the characteristic of the human visual system. Let $V = (v_{k,l})$ denote a Gaussian filter, i.e. a 2-dimensional symmetric matrix of size $(2w+1)\times(2w+1)$, where each non-negative real number $v_{k,l}$ $(-w \leq k,l \leq w)$ is determined by a 2-dimensional Gaussian distribution such that their sum is 1. In other words,

$$v_{k,l} = c \cdot e^{-\frac{k^2+l^2}{2\sigma^2}} \quad (1)$$

where σ is a parameter of the Gaussian distribution and c is a fixed real number to satisfy $\sum_{-w\leq k,l\leq w} v_{k,l} = 1$. Let $R = (r_{i,j})$ be the projected gray scale image of a binary image $B = (b_{i,j})$ obtained by applying the Gaussian filter as follows:

$$r_{i,j} = \sum_{-w\leq k,l\leq w} v_{k,l} b_{i+k,j+l} \quad (0 \leq i,j \leq n-1) \quad (2)$$

Clearly, from $\sum_{-w\leq k,l\leq w} v_{k,l} = 1$ and $v_{k,l}$ is non-negative, each $r_{i,j}$ takes a real number in the range $[0,1]$ and thus, the projected image R is a gray scale image. We can say that a binary image B is a good approximation of original image A if the difference between A and R is small enough. Hence, we define the error of B as follows. The error $e_{i,j}$ at each pixel location (i,j) is defined by

$$e_{i,j} = (a_{i,j} - r_{i,j})^2 \quad (3)$$

and the total error is defined by

$$Error(A,B) = \sum_{0\leq i,j\leq n-1} e_{i,j}. \quad (4)$$

Since the Gaussian filter approximates the characteristics of the human visual system, we can think that image B reproduces original gray scale image A if $Error(A,B)$ is small enough. The best binary image that reproduces A is a binary image B which is given by the following formula:

$$B^* = \arg\min_B Error(A,B). \quad (5)$$

It is very hard to find the optimal binary image B^* for a given gray scale image A. The idea of the

[†]For simplicity, we assume that images are square.

DBS is to find a near optimal binary image B such that $Error(A, B)$ is sufficiently small. For this purpose, the DBS repeats improvement of binary image B. The value of a particular pixel $b_{i,j}$ is modified by the following two operations:

Toggling This operation is to toggle the value of $b_{i,j}$, that is, $b_{i,j} \leftrightarrow 1 - b_{i,j}$. The value of $b_{i,j}$ is toggled if $Error(A, B)$ decreases.

Swapping Let $b_{i',j'}$ be a neighbor pixel of $b_{i,j}$, that is, both $|i - i'| \leq 1$ and $|j - j'| \leq 1$ hold. This operation is to exchange the values of $b_{i,j}$ and $b_{i',j'}$, that is $b_{i,j} \leftrightarrow b_{i',j'}$. Swapping operation is performed if $Error(A, B)$ decreases.

Clearly, toggling and swapping operations do not increase the error and improve the binary image B. In the DBS, these operations are executed in the raster order. Further, this raster order improvement is repeated until no more improvement by toggling or swapping operations is possible.

3. Clippings of binary images generated by the DBS

Although the DBS generates high-quality binary images, it does not work well in highlights and shadows. It has *clippings*, that is, the highlight and the shadow areas have no dots and lose the tone of the original image. Figure 3 shows the resulting binary images by the DBS. The left shadow area has no white dots and the tone is lost. Similarly, in the right highlight area, black dots disappear. Figure 4 shows the resulting binary image generated by the DBS. Black dots in the woman's face are lost and pseudo borders appear. Also, white dots in her hair are removed. The main purpose of this section is to show the reason why binary images generated by the DBS have clippings and clarify the range of intensities that may have clippings.

As before, let $V = (v_{k,l})$ denote a Gaussian filter of size $(2w+1) \times (2w+1)$. Let A_d denote a flat tone gray scale image of size $(2w+1) \times (2w+1)$ such that $a_{i,j} = d$ $(0 \leq i, j \leq 2w)$ for an intensity level d. Also, let B be a initial binary image with each pixel taking value 0. Suppose that the center pixel $b_{w,w}$ of B is toggled, and let B' be the resulting binary image. Let $e(d)$ be the improvement of B' over B in terms of the error for A, that is,

$$\begin{aligned} e(d) &= Error(A_d, B') - Error(A_d, B) \\ &= \sum_{-w \leq k,l \leq w} (d - v_{k,l})^2 - \sum_{-w \leq k,l \leq w} d^2 \\ &= \sum_{-w \leq k,l \leq w} v_{k,l}^2 - 2d \end{aligned}$$

If d is much smaller than the minimum of $v_{k,l}$, then

$$e(d) = \sum_{-w \leq k,l \leq w} v_{k,l}^2 > 0$$

On the other hand, if d is larger than the maximum of $v_{k,l}$, then

$$e(d) = -2d < 0$$

Thus, there exists a real number D such that $e(d) > 0$ for all $d < D$, and $e(D) = 0$. For d $(d < D)$, the toggling operation does not decrease the error. Therefore, B is the best binary image with the minimum error, although it has no white pixel. It follows that shadows consisting of pixels with intensity smaller than D have no white pixel in the corresponding areas of the binary image if we use the DBS. By the same reason, highlights with intensity larger than $1 - D$ have no black pixel. Table 1 shows the resulting binary images for $A_{\frac{1}{255}}$, $A_{\frac{2}{255}}$, ..., $A_{\frac{7}{255}}$ using the DBS. Since the DBS with Gaussian parameter $\sigma = 1.2$ generates good quality binary images[4], [9], [10], we have also selected $\sigma = 1.2$. If $\sigma = 1.2$, $v_{3,3} \approx \frac{c}{518.0}$ and $v_{4,0} \approx \frac{c}{258.7}$, while $v_{0,0} = c$. Thus, since we are assumes 8-bit gray scale images and the minimum intensity level is $\frac{1}{255}$, $w = 3$ is large enough for $\sigma = 1.2$. Also, we have performed a lot of experiments on different values of w, and found when w is larger than 3 the DBS works almost the same as $w = 3$. The resulting images for $A_{\frac{1}{255}}$, $A_{\frac{2}{255}}$, ..., $A_{\frac{5}{255}}$ have no white dots. It follows that, D is between $\frac{5}{255}$ and $\frac{6}{255}$ if $\sigma = 1.2$ and $w = 3$.

Let us analyze the relation between the threshold value D and the parameter σ of the Gaussian filter. Let $e(d) = 0$, then we can get $D(\sigma)$ as follows:

$$\begin{aligned} D(\sigma) &= \frac{1}{2} \sum_{-w \leq k,l \leq w} v_{k,l}^2 \\ &= \frac{1}{2} \sum_{-w \leq k,l \leq w} c^2 \cdot e^{-\frac{k^2+l^2}{\sigma^2}}. \\ c &= \frac{1}{\sum_{-w \leq k,l \leq w} e^{-\frac{k^2+l^2}{2\sigma^2}}}. \end{aligned}$$

To make the observation of $D(\sigma)$ more easily, the discrete integral is converted into the continuous integral as follows:

$$\begin{aligned} \frac{1}{c} &= \int_{-w}^{w} \int_{-w}^{w} e^{-\frac{x^2+y^2}{2\sigma^2}} dxdy \\ &= \int_{0}^{2\pi} d\theta \int_{0}^{\sqrt{2}w} e^{-\frac{r^2}{2\sigma^2}} rdr \\ &= 2\pi\sigma^2(1 - e^{-\frac{2w^2}{2\sigma^2}}). \\ D(\sigma) &= \frac{c^2}{2} \int_{-w}^{w} \int_{-w}^{w} e^{-\frac{x^2+y^2}{\sigma^2}} dxdy \\ &= \frac{c^2}{2} \int_{0}^{2\pi} d\theta \int_{0}^{\sqrt{2}w} e^{-\frac{r^2}{\sigma^2}} rdr \\ &= \frac{c^2}{2} \cdot \pi\sigma^2(1 - e^{-\frac{2w^2}{\sigma^2}}). \end{aligned}$$

Substitute for c, the expression of $D(\sigma)$ is:

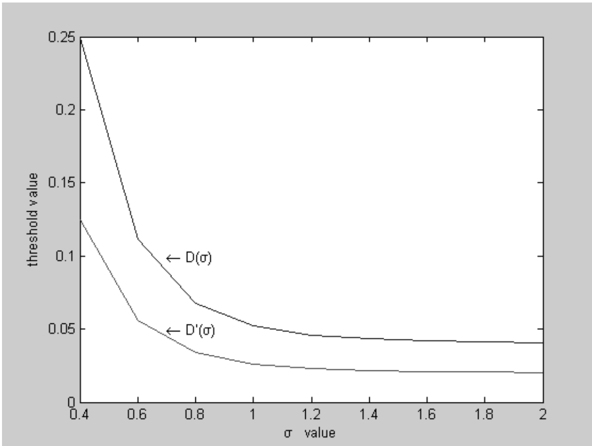


Fig. 1 The distribution of D depending on σ when $w = 3$ and the distribution of D' depending on σ when $L = 3$ and $w = 3$

$$D(\sigma) = \frac{1}{8\pi} \cdot \frac{1 + e^{-\frac{w^2}{\sigma^2}}}{\sigma^2(1 - e - \frac{w^2}{\sigma^2})}. \tag{6}$$

The relation of D to σ when $w = 3$ is shown in Figure 1. In the figure, we can see D decreases as σ increases. When w is larger than 3, the curve of $D(\sigma)$ doesn't have perceptible difference from that as $w = 3$.

4. Our clipping-free DBS-based halftoning

This section is devoted to show our clipping-free DBS. The key idea of our DBS-based halftoning method is to use the Ordered Dither for the pixels with intensity smaller than D or larger than $1 - D$ and then use the DBS.

We first determine a threshold array $T = (t_{i,j})$ of size $m \times m$ used for shadows. Since this T is used for the pixel values no more than D, it is not necessary to determine threshold value larger than D. Thus, for each i $(0 \le i \le D)$, $\frac{m^2}{255}$ elements in T takes value $\frac{i}{255}$. For highlight pixel with intensity larger than $1 - D$, we can use a threshold array $T' = (t'_{i,j})$ such that $t'_{i,j} = 1 - t_{i,j}$.

The goal of determining T is to distribute the threshold values in T uniformly. The uniformity is defined as follows. Let $u(i,j)$ denote the Euclidean distance to a closest threshold value no more than $t_{i,j}$. In other words,

$$u(i,j) = \min\{\sqrt{(i-i')^2 + (j-j')^2} \mid t_{i',j'} \le t_{i,j}\}.$$

The uniformity $u(T)$ of T is the sum of $u_{i,j}$, that is,

$$u(T) = \sum_{0 \le i,j \le m-1} u(i,j).$$

Clearly, if threshold value distributed more uniformly, the uniformity $u(T)$ is larger.

We first assign threshold value $\frac{0}{255}$ to $\frac{m^2}{255}$ elements in T. For this purpose, we select $\frac{m^2}{255}$ elements in T at random and assign $\frac{0}{255}$ to them. After that, we move each $\frac{0}{255}$ to a neighbor element if the uniformity $u(T)$ increases. The reader should have no difficulty to confirm that, this operation is very similar to swapping operation of the DBS. This swapping operation is repeated until no more improvement on $u(T)$ is possible. Next, we assign threshold value $\frac{1}{255}$ to $\frac{m^2}{255}$ elements in T. Similarly, we select $\frac{m^2}{255}$ elements that were not assigned $\frac{0}{255}$ and assign $\frac{1}{255}$ to them. After that, the swapping operation is performed for these elements with threshold value $\frac{1}{255}$. The same procedure is repeated until threshold values from $\frac{0}{255}$ to D are determined. If T is small, the generated binary image may have periodic artifact with frequency $m \times m$ pixels. Thus, T should be as large as possible. In our experiment we found a threshold array of size 512×512 is large enough. Table 1 shows the resulting binary image obtained by using the threshold array T thus obtained. In each binary image, white pixels are uniformly distributed. Also, please note that, it is easy to generate a level threshold array with deeper depth using this method. For example, we can generate threshold array with values $\frac{0}{1024}, \frac{1}{1024}, \frac{2}{1024}, \ldots$, if they are necessary for dot-gain calibration.

We are now in position to explain our clipping-free DBS. Suppose that a gray scale image $A = (a_{i,j})$ to be halftoned is given. We first apply the threshold array T to pixels $a_{i,j}$ of A such that $a_{i,j} < D$ or $a_{i,j} > 1-D$, and obtain a binary image $B = (b_{i,j})$. Next, we assign label *determined/undetermined* to every pixel as follows:

> $b_{i,j}$ is *determined*, if $(a_{i,j} < D$ and $b_{i,j} = 1)$ or $(a_{i,j} > 1 - D$ and $b_{i,j} = 0)$, and
> $b_{i,j}$ is *undetermined*, otherwise.

In other words, if $b_{i,j}$ is minority pixel in shadows or in highlights, then it is a determined pixel. After that, the DBS is executed for all undetermined pixels, that is, toggling and swapping operations repeated in the raster scan order until no more improvement of the error is possible.

To obtain these images, we have generated a threshold array of size 512×512. We also use the Gaussian filter with parameter $\sigma = 1.2$ and $w = 3$, and the threshold value $D = \frac{6}{255}$ is used. Figures 3(100×600 pixels) and 4(600×270 pixels) show the resulting binary images. We can see clearly the original tone is preserved in shadows and highlights. The resulting images look smooth and contain more variety of patterns. They take on good visual texture. In Figure 2, we also show the DBS tone reproduction curve (TRC) [20] and our clipping-free DBS tone reproduction curve, which show the relation between a sequence of constant valued gray scale images and the mean value of the corresponding binary images. From the curve, we can see our halftoning method reproduces the tone correctly.

Table 1 The resulting binary images for flat tone images

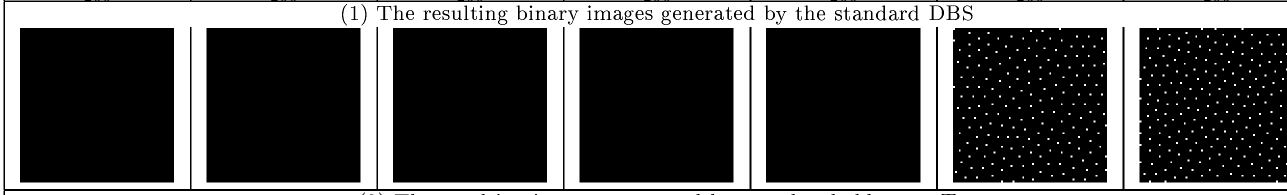
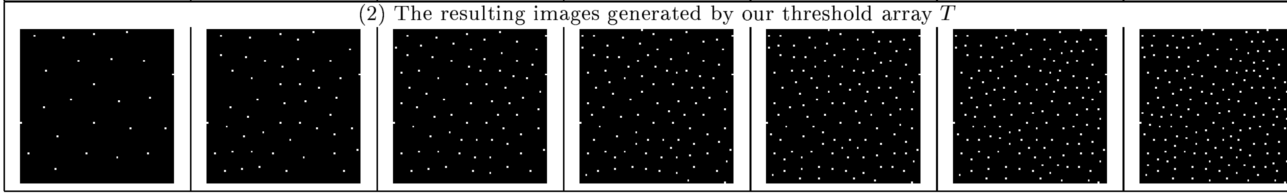
Intensity levels of the original flat gray scale image						
$\frac{1}{255}$	$\frac{2}{255}$	$\frac{3}{255}$	$\frac{4}{255}$	$\frac{5}{255}$	$\frac{6}{255}$	$\frac{7}{255}$
(1) The resulting binary images generated by the standard DBS						
						
(2) The resulting images generated by our threshold array T						
						

Table 2 shows the computing time and the number of toggling/swapping operations to obtain the binary images. We have used Pentium4 (3.8GHz)-based Linux (CentOS5.1) PC and gcc with -O2 option to evaluate the computing time. From the table, we can see that standard DBS and our clipping-free DBS take almost the same time to obtain the binary image.

5. Multitoning using a threshold array and the DBS-based technique

The main purpose of this section is to show how we use the threshold array and the DBS to obtain a multitone image.

Recall that, the process of multitoning is to generate, for a given gray scale image $A = (a_{i,j})$, an L-level multitone image $M = (m_{i,j})$ with each pixel taking a value in $\{\frac{0}{L-1}, \frac{1}{L-1}, \ldots, \frac{L-1}{L-1}\}$. The key idea to obtain M is to round each pixel of A. That is, $m_{i,j}$ takes a value either $\frac{\lfloor a_{i,j}(L-1) \rfloor}{L-1}$ (round down) or $\frac{\lceil a_{i,j}(L-1) \rceil}{L-1}$ (round up). For the purpose of determining if "round down" or "round up", we use binary image $B = (b_{i,j})$ obtained by halftoning, such that "round down" if $b_{i,j} = 0$ and "round up" if $b_{i,j} = 1$.

Using the idea above, it is not difficult to see how a multitone image is obtained by a threshold array $T = (t_{i,j})$ used for halftoning as follows. Recall that each $t_{i,j}$ takes one of the real numbers $\frac{0}{255}, \frac{1}{255}, \ldots, \frac{254}{255}$. First, for a given gray scale image A, we compute a binary image B using the following formula †

$$b_{i,j} = \begin{cases} 0 & \text{if } \mathrm{frac}(a_{i,j} \cdot (L-1)) \leq t_{i \bmod m, j \bmod m} \\ 1 & \text{if } \mathrm{frac}(a_{i,j} \cdot (L-1)) > t_{i \bmod m, j \bmod m} \end{cases}$$

After that, every pixel $m_{i,j}$ of an L-level multitone image M is determined as follows:

$$m_{i,j} = \begin{cases} \frac{\lfloor (a_{i,j} \cdot (L-1)) \rfloor}{L-1} & \text{if } b_{i,j} = 0 \\ \frac{\lceil (a_{i,j} \cdot (L-1)) \rceil}{L-1} & \text{if } b_{i,j} = 1 \end{cases} \quad (7)$$

† "frac" denotes the value removing the integer part of the argument.

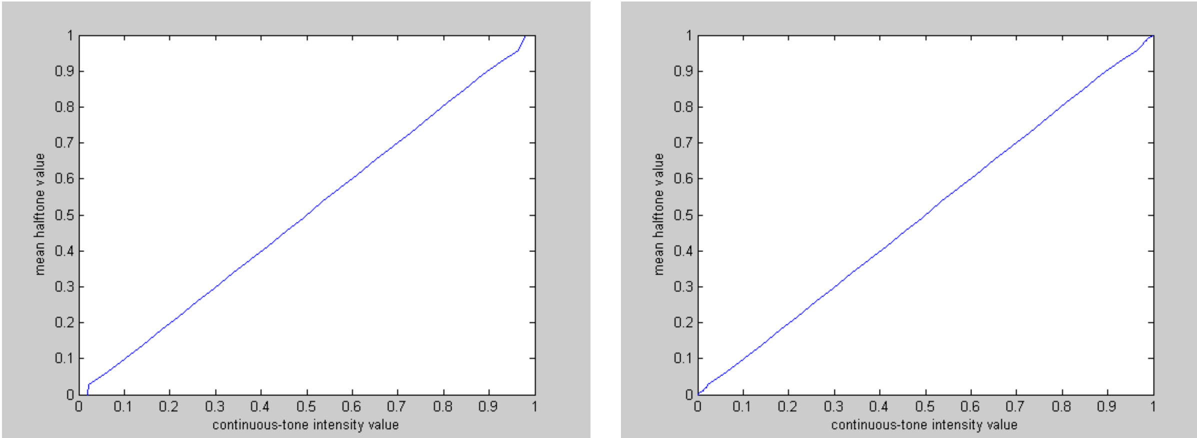
An L-level multitone image can be obtained using the DBS as follows. We compute a binary image that determines rounding up or rounding down. Let B be the current binary image thus obtained by the DBS. The corresponding L-level multitone image M can be computed in a similar way to Eq. (7). We then compute the projected image R of the L-level multitone image M using the Eq. (2) and the total error of M with respect to the original gray scale image A using the Eq. (3) and (4). Similarly to the DBS for halftoning, we repeatedly execute toggling and swapping operations for B to find a near optimal multitone image M with small total error. These operations are repeated until no more improvement is possible.

If we use this DBS-based multitoning method to obtain an L-level multitone image, the resulting image has no tone with intensity levels close to $\frac{0}{L-1}, \frac{1}{L-1}, \ldots, \frac{L-1}{L-1}$. Figure 3 shows the resulting 3-level multitone image. It has clippings with intensity levels close to $\frac{0}{2}$, $\frac{1}{2}$, and $\frac{2}{2}$. Figure 5 shows the multitone image for a sky image of size 600×270. The sky has clipping areas that have no dots with intensity $\frac{0}{2}$ or $\frac{1}{2}$. The continuous tone of the sky is lost and pseudo border lines appear. White dots in the shadow areas of the building are removed.

Let us evaluate the threshold value D' that determine the areas which have no dot using the DBS-based multitoning for obtaining an L-level multitone image. Again, let $V = v_{k,l}$ denote a Gaussian filter of size $(2w+1) \times (2w+1)$. For a fixed real number $d \geq 0$, let $v'_{k,l} = d - \frac{v_{k,l}}{L-1}$. Let $e'(d)$ be the function such that

$$\begin{aligned} e'(d) &= \sum_{-w \leq k,l \leq w} {v'_{k,l}}^2 - \sum_{-w \leq k,l \leq w} d^2 \\ &= \sum_{-w \leq k,l \leq w} \left(\frac{v_{k,l}}{L-1}\right)^2 - \frac{2d}{L-1}. \end{aligned}$$

By the same argument of the DBS for halftoning, we can find a threshold value D' such that $e(D') = 0$ and $e'(d) > 0$ for all $d < D'$. For such D', the resulting multitone image has no dots for the areas with intensity



(1) DBS tone reproduction curve (2) Our clipping-free DBS tone reproduction curve

Fig. 2 Tone Reproduction Curve

Table 2 Computing time and number of operations for the woman image of size 600 × 270

		Computing time	# of operations
halftoning	standard DBS	16.0s	4786586
	our clipping-free DBS	16.1s	4936514
multitoning	standard DBS	6.21s	2941972
	our clipping-free DBS	5.83s	2869420

levels below D' and above $1-D'$. Further, the resulting image has no tone for the areas with intensity levels in the ranges $[\frac{1}{L-1}-D', \frac{1}{L-1}+D']$, $[\frac{2}{L-1}-D', \frac{2}{L-1}+D']$, $\ldots$, $[\frac{L-1}{L-1}-D', \frac{L-1}{L-1}+D']$. Figure 1 also shows the relation of D' and σ.

6. Our clipping-free DBS for multitoning

This section is devoted to show our clipping-free DBS multitoning algorithm.

Similarly to our clipping-free DBS halftoning, we first determine the binary image using our threshold array T computed in Section 4, and then assign determined/undetermined labels to every pixel. For every undetermined pixels we use the multitoning version of the DBS. The details are spelled out as follows.

Suppose that a gray scale image $A=(a_{i,j})$ to be multitoned is given. As before, the threshold value D' such that $e(D')=0$ by formula 6. We first use the threshold array T to A and obtain the binary image $B=b_{i,j}$. Recall that the multitone image can be obtained using the binary image B. Next, we assign label *determined/undetermined* to every pixel as follows:

> $b_{i,j}$ is *determined*, if $(frac(a_{i,j}\cdot(L-1))<D'$ and $b_{i,j}=1)$ or $(frac(a_{i,j}\cdot(L-1))>(1-D')$ and $b_{i,j}=0)$, and
> $b_{i,j}$ is *undetermined*, otherwise.

In other words, $b_{i,j}$ is undetermined if it is minority, that is, if it is rounding down and $a_{i,j}$ is in $[\frac{0}{L-1}, \frac{0}{L-1}+D']$, $[\frac{1}{L-1}, \frac{1}{L-1}+D']$, $\ldots$, $[\frac{L-2}{L-1}, \frac{L-2}{L-1}+D']$. It is also undetermined if it is rounding up and $a_{i,j}$ is in $[\frac{1}{L-1}-D', \frac{1}{L-1}]$, $[\frac{2}{L-1}-D', \frac{2}{L-1}]$, $\ldots$, $[\frac{L-1}{L-1}-D', \frac{L-1}{L-1}]$. Next, the DBS is executed for all undetermined pixels, that is, toggling and swapping operations repeated in the raster order until no more improvement on the total error is possible.

The 3-level multitoning results using our clipping-free DBS are shown in Figures 3 and 5. We use the Gaussian filter with parameter $\sigma=1.2$ and $w=3$, and the threshold value $D'=\frac{4}{255}$. We can see that they do not have clippings, look very smooth and contain more variety of patterns. They show very good visual texture.

Table 2 also shows the computing time and the number of toggling/swapping operations to obtain the multitone images. We can see that standard DBS and our clipping-free DBS take almost the same time for the multitoning.

7. Conclusions

In this paper, we first show the reason why binary images generated standard DBS (Direct Binary Search) have clippings in highlights and shadows, and clarify area of binary images that have clippings. We then go on to presented clipping-free halftoning method based on the DBS. The key idea of our method is to use the Orderd Dither method in highlights and shadows that may have clippings, and apply DBS to the other areas of the image. The dither matrix used by the Orderd Dither method is generated by DBS-based algorithm using Euclidean distance metric, and thus, the dots in highlights and shadows are uniformly distributed. Finally, our halftoning method is extended to generate a clipping-free multitone images. The basic idea to use DBS-based method for multitoning such that "rounding up" or "rounding down" at each pixel is modified by toggling/swapping of the DBS. From the experimental results, binary and multitone image generated by our DBS-based method show very good visual textures.

References

[1] M. Analoui and J.P. Allebach. "Model-based halftoning by direct binary search", In *Proc. SPIE/IS&T Symposium on Electronic Imaging Science and Technology*, volume 1666, pages 96–108, 1992.

[2] T. Asano, "Digital Halftoning: Algorithm Engineering Challenges", IEICE Transactions on Information and Systems, pp. 159-178, 2003.

[3] D. J. Lieberman and J.P. Allebach, "Efficient model based halftoning using direct binary search", Proc. of International Conference on Image Processing, Vol. 1, pp. 775 – 778, 1997.

[4] F.A. Baqai and J. P. Allebach, "Halftoning via direct binary search using analytical and stochastic printer models", IEEE Transactions on Image Processing, Vol. 12, No. 1, pp. 1 – 15, 2003.

[5] B.E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures", IEEE International Conference on Communications, vol.26, pp.11-15,1973.

[6] C. Chang, "Photo quality color printing by using light black ink", United States Patent 6765693, 2004.

[7] R.W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale", SID 75 Digest, Society for Information Display, pp.36-37, 1975.

[8] D.E. Knuth, "Digital halftones by dot diffusion", CAM Trans. Graphics, vol.6, no.4, pp.245-273, 1987.

[9] Y. Ito, and K. Nakano, "FM Screening by the Local Exhaustive Search with Hardware Acceleration", International Journal on Foundations of Computer Science, Vo.16, No.1, pp. 89-104, Feb. 2005

[10] Y. Ito and K. Nakano, "Cluster-dot Screening by Local Exhaustive Search with Hardware Acceleration", Proc. of Workshop on Advances in Parallel and Distributed Computational Models(CD-ROM), 2007.

[11] D. L. Lau and G. R. Arce. *Modern Digital Halftoning*. Marcel Dekker, 2001.

[12] K. Nakano, "Various Screening Methods", Convertech, Vol. 36, No.1, pp.72–77, 2008.

[13] T. Okuda, and K. Ohshima, "The High-Speed Business Printer with a Line Type Inkjet Print Head", The Journal of the Society of Photographic Science and Technology of Japan, Vol.68, No.1, pp.26-41, 2005.

[14] V. Ostromoukhov, P. Emmel, N. Rudaz, I. Amidror, R.D. Hersch, "Multi-level Colour Halftoning Algorithms", Int. Symposium on Advanced Imaging and Network Technologies, SPIE Vol. 2949, 332-340, 1996.

[15] R. Uichney. "The void-and-cluster method for dither array generation", Proceedings SPIE, Human Vision, Visual Processing, Digital Displays IV, vol.1913,pp.332-343,1993.

[16] Q. Yu, K. J. Parker, K.E. Spaulding, and R. L. Miller, "Digital multitoning with overmodulation for smooth texture transition", J. Electronic Imaging, Vol. 8, No. 3, pp. 311-321, 1999.

[17] D. A. Carrara, M. Analoui, and J. P. Allebach, "Recent progress in digital halftoning", in *Proc. Imaging Science and Technology 8th Int. Congr. Adv. Non-Impact Printing Tech.*, Williamsburg, VA, Oct. 1992, pp. 265-270.

[18] D. J. Lieberman and J. P. Allebach, "Digital halftoning using the direct binary search algorithm", in *Proc. 1996 1st Int. Conf. High Technology*, Chiba, Japan, Sept. 1996, pp. 114-124.

[19] Sang Ho Kim and Jan P. Allebach, "Impact of HVS Models on Model-Based Halftoning", IEEE Transactions on Image Processing, Vol. 11, pp. 258-269, 2002. "A New Hybrid Multitoning Based on the Direct Binary Search", Prof. of International Multi Conference of Engineers and Computer Scientists, Vol. 1, pp. 627-632, March 2008.

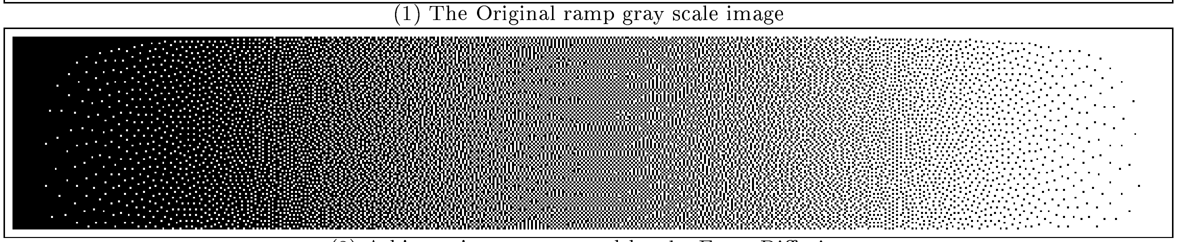
[20] David J. Lieberman and Jan P. Allebach, "A Dual Interpretation for Direct Binary Search and Its Implications for Tone Reproduction and Texture Quality", IEEE Transactions on Image Processing, Vol. 9, No. 11, Nov. 2000, pp. 1950-1963.

Xia Zhuge received the BE from Department of Communication Engineering, Xi'an University of Science and Technology, China in 2003; and received the ME from Department of Technical Physics, Xidian University, China in 2007. Now she is a doctor course student at School of Engineering, Hiroshima University, Japan. Her research interest is image processing.

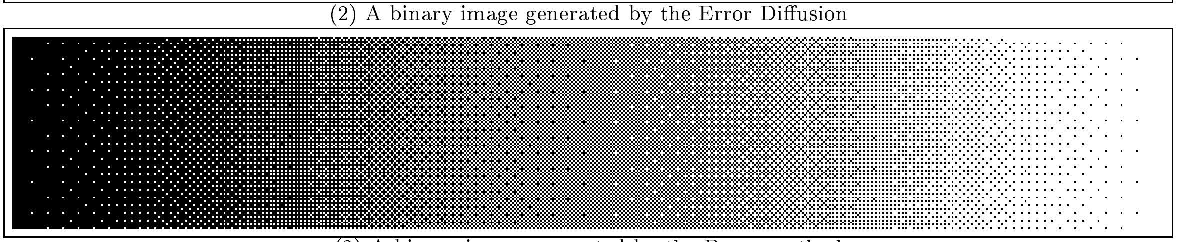
Koji Nakano received the BE, ME and Ph.D degrees from Department of Computer Science, Osaka University, Japan in 1987, 1989, and 1992 respectively. In 1992-1995, he was a Research Scientist at Advanced Research Laboratory. Hitachi Ltd. In 1995, he joined Department of Electrical and Computer Engineering, Nagoya Institute of Technology. In 2001, he moved to School of Information Science, Japan Advanced Institute of Science and Technology, where he was an associate professor. He has been a full professor at School of Engineering, Hiroshima University from 2003. He has published extensively in journals, conference proceedings, and book chapters. He served on the editorial board of journals including IEEE Transactions on Parallel and Distributed Systems, IEICE Transactions on Information and Systems, and International Journal of Foundations on Computer Science. He has also guest-edited several special issues including IEEE TPDS Special issue on Wireless Networks and Mobile Computing, IJFCS special issue on Graph Algorithms and Applications, and IEICE Transactions special issue on Foundations of Computer Science. He has organized conferences and workshops including International Conference on Parallel and Distributed Computing, Applications and Technologies, IPDPS Workshop on Advances in Parallel and Distributed Computational Models, and ICPP Workshop on Wireless Networks and Mobile Computing. His research interests includes image processing, hardware algorithms, FPGA-based reconfigurable computing, parallel computing, mobile computing, algorithms and

architectures.

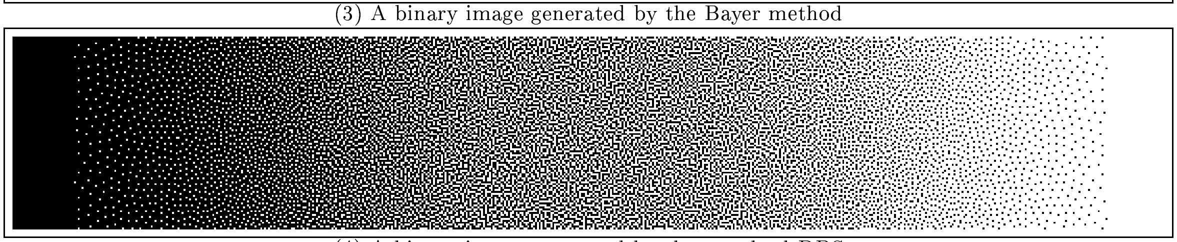
(1) The Original ramp gray scale image



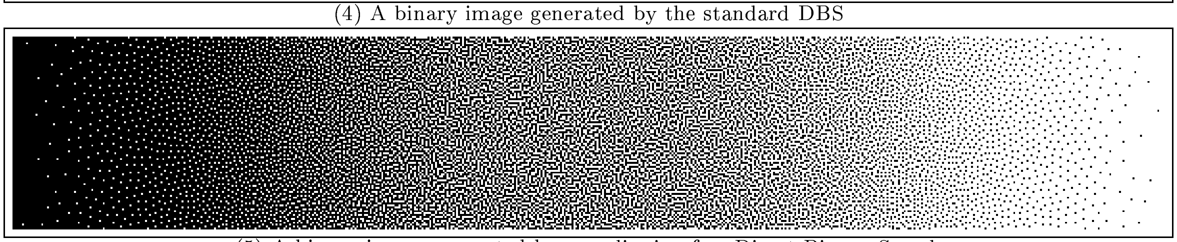
(2) A binary image generated by the Error Diffusion



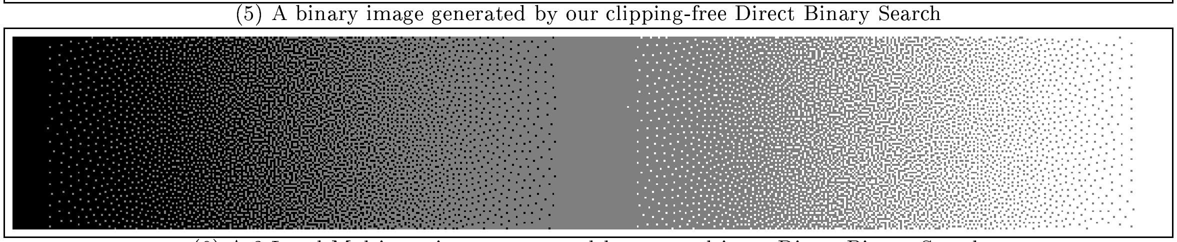
(3) A binary image generated by the Bayer method



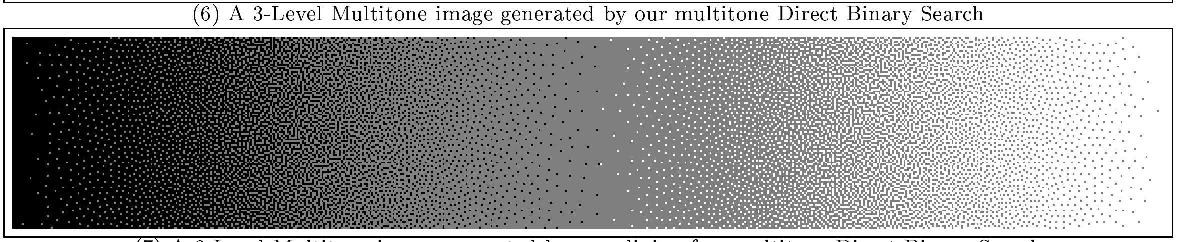
(4) A binary image generated by the standard DBS



(5) A binary image generated by our clipping-free Direct Binary Search



(6) A 3-Level Multitone image generated by our multitone Direct Binary Search

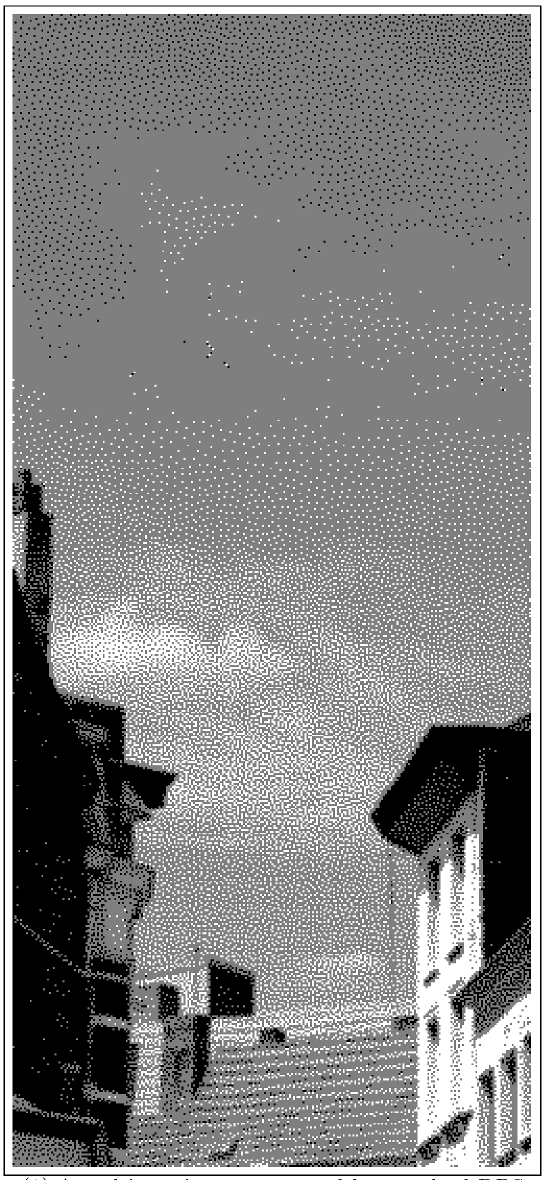


(7) A 3-Level Multitone image generated by our cliping-free multitone Direct Binary Search

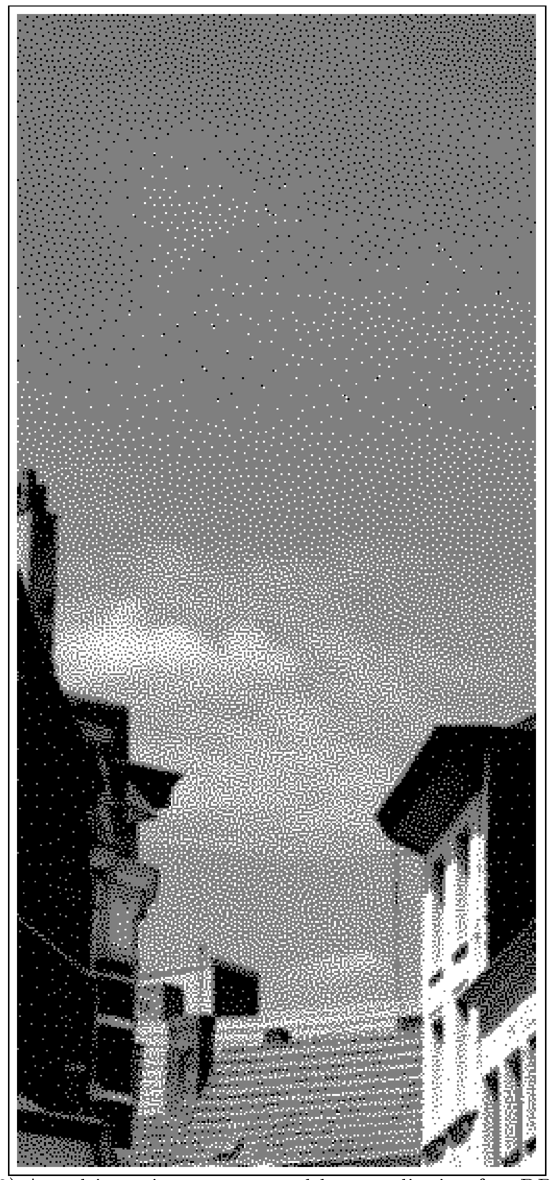
Fig. 3 The resulting halftone and multitone images for the ramp image

(1) A binary image generated by the standard DBS (2) A binary image genearted by our clipping-free DBS

Fig. 4 The resulting binary image for a woman image



(1) A multitone image generated by standard DBS



(2) A multitone image genearted by our clipping-free DBS

Fig. 5 The resulting 3-level multitone images for a sky image