

Juxtaposed Color Halftoning Relying on Discrete Lines

Vahid Babaei, and Roger D. Hersch, *Member, IEEE*

Abstract—Most halftoning techniques allow screen dots to overlap. They rely on the assumption that the inks are transparent, i.e. the inks do not scatter a significant portion of the light back to the air. However, many special effect inks such as metallic inks, iridescent inks or pigmented inks are not transparent. In order to create halftone images, halftone dots formed by such inks should be juxtaposed, i.e. printed side by side. We propose an efficient juxtaposed color halftoning technique for placing any desired number of colorant layers side by side without overlapping. The method uses a monochrome library of screen elements made of discrete lines having rational thicknesses. Discrete line juxtaposed color halftoning is performed efficiently by multiple accesses to the screen element library.

Index Terms—Color halftoning, discrete lines, juxtaposed halftoning, Neugebauer primaries, opaque inks

I. INTRODUCTION

HALFTONING algorithms try to reproduce the visual impression of a continuous tone image by taking advantage of the low-pass filtering property of the human visual system (HVS). In classical color halftoning algorithms, such as clustered dot and blue noise dithering, a halftone layer is created for each ink separately [1]. The final color halftone image is formed by the superposition of all the layers. The screen dot layers form partially overlapping screen dots. Overlapped screen dots form new colorants¹ under the assumption that the inks are transparent, i.e. they do not scatter light back to the surface. There are however applications with strongly scattering inks such as opaque inks, metallic inks, or inks providing special effects such as daylight fluorescent inks and iridescent inks. In such applications, in order to obtain predictable halftone colors, one needs to print the different colorant halftone dots side by side without overlapping.

Previous attempts for side by side printing of colorants comprise Kueppers approach of 7 color printing [2], error diffusion in color space [3]-[5], multi-color dithering [6], juxtaposed halftoning using screen libraries [7] and error

diffusion of Neugebauer primaries [8], [9]. Ostromoukhov and Hersch [6] presented a juxtaposed multi-color dithering technique where amounts of colorants are converted into dither value intervals. The resulting colorant surfaces form colorant rings that follow the level lines of the dither function. By using a space filling curve [10] as dither function, partially clustered juxtaposed screen shapes can be generated [11]. Within the framework of fluorescent imaging, Hersch *et al.* [7] created a new clustered-dot juxtaposed halftoning algorithm for printing images with fluorescent inks. The resulting halftone screens allow three colorants to be printed side by side within a given screen tile. Morovič *et al.* [8], [9] proposed an approach for printing with freely chosen amounts of Neugebauer primaries by relying on error diffusion halftoning. Color separation is performed by segmenting the color space into convex subspaces (e.g. tetrahedra) whose vertices are formed by the Neugebauer primaries.

Ideally, juxtaposed halftoning should have similar properties as conventional halftoning. It should provide the possibility of printing with a sufficient number of colorants and tone variations. It should also provide some clustering behavior, be able to reproduce image details at a frequency higher than the screen frequency, exhibit as least artifacts as possible and offer support for an efficient implementation [12].

In this paper, we introduce a new juxtaposed halftoning algorithm which creates side by side laid out colorant halftone lines without limitation in the number of colorants. The proposed method relies on discrete line geometry which provides subpixel precision for creating discrete parallelograms. The screen elements are formed by parallelograms made of discrete line segments whose relative subpixel thicknesses are set according to the desired colorant surface coverages. The parallelogram screen elements form a library comprising all possible discrete line thickness variations. The final color halftone screen is created by accessing and combining binary screen elements stored within the library. In Section II, we introduce the discrete line which is the building block of our juxtaposed halftoning algorithm. In Section III, we describe the discrete line drawing algorithm.

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This work was supported in part by the Swiss National Science Foundation under Grant 200020_126757/1. The authors are with the Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland (e-mail: vahid.babaei@epfl.ch; rd.hersch@epfl.ch).

¹ We use the term “colorant” for unprinted paper, solid inks and the superposition of solid inks printed on paper. Classical halftones made with cyan, magenta and yellow inks comprise 8 colorants, also called Neugebauer primaries: paper white, cyan, magenta, yellow, blue as the superposition of cyan and magenta, green as the superposition of cyan and yellow, red as the superposition of magenta and yellow, and black as the superposition of cyan, magenta and yellow.

Section IV outlines the procedure of creating bilevel screen elements. Multi-colorant juxtaposed halftoning and its efficient implementation are presented in Section V. In Section VI, we discuss the parameters influencing the properties of the resulting halftones. Finally, in Section VII we show experimental results.

II. DISCRETE LINES

The arithmetic definition of a discrete line introduced by Reveillès is a fundamental notion in digital geometry [13]-[15]. It allows mastering the creation of discrete lines of subpixel thicknesses. Since it is based on rational numbers, the discrete line plotting function does not propagate errors as is the case with floating point algorithms. A set D of points (x, y) in \mathbb{Z}^2 belongs to the discrete line if and only if each member of this set satisfies

$$\gamma \leq ax - by < \gamma + w \quad (1)$$

In other words

$$D(a, b, \gamma, w) = \{(x, y) \in \mathbb{Z}^2 \mid \gamma \leq ax - by < \gamma + w\} \quad (2)$$

where parameters a , b , γ and w are integers, a/b is the line slope, γ defines the *affine offset* indicating the line position in the plane and w determines its thickness.

In this work, pixels are represented by unit squares centered on integer points. A discrete line with $0 < |a| < |b|$ has two *Euclidean* support lines. The superior support line is given by

$$y_{\text{sup}} = \frac{a}{b}x - \frac{\gamma}{b} \quad (3)$$

and the inferior support line is given by

$$y_{\text{inf}} = \frac{a}{b}x - \frac{\gamma + w}{b} \quad (4)$$

By subtracting these two equations, we obtain *vertical thickness* w/b which is the vertical distance between the superior and inferior support lines. The arithmetic thickness parameter w controls the vertical thickness and the connectivity of the line:

- If $w < |b|$, the line is disconnected and we call it *thin line*,
- If $w = |b|$, the line is strictly 8-connected; it is called *naive digital line* and has exactly the vertical thickness of 1,
- If $w > |b|$, the line is a *thick line* with thickness greater than 1.

Fig. 1 shows discrete lines having different thicknesses.

Another interesting property of a discrete line is its *b-periodicity*. As shown in Fig. 2, for a given naive digital line in the first octant with parameters a and b , after b pixels in the horizontal direction, the same line segment is repeated. Therefore, the discrete line is invariant under the translation $k[b \ a]^T$, for any integer k . The main advantage of b -periodicity is that we can limit our study to pixels $x \in [0, b-1]$. Furthermore, we can use this property for efficient discrete line plotting.

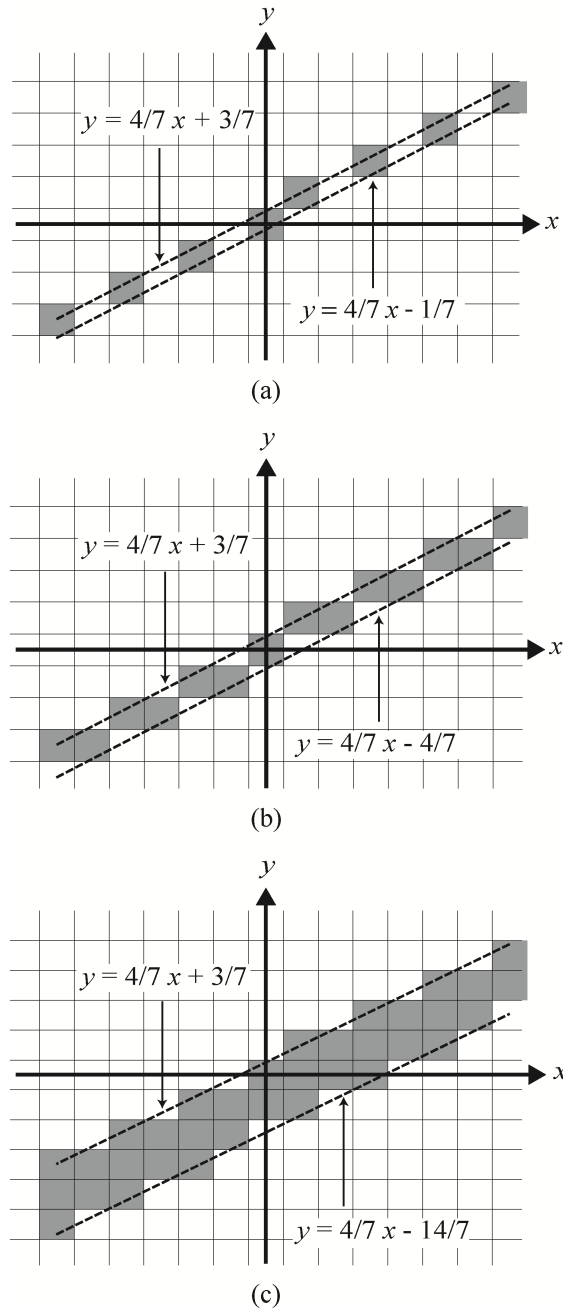


Fig. 1. Discrete lines with $a = 4$, $b = 7$ and $\gamma = -3$ having different thicknesses: (a) thin $w = 4$, (b) naive $w = 7$, and (c) thick $w = 17$.

III. DISCRETE LINE PLOTTING

The first step towards discrete line halftoning is the ability of generating discrete lines with any desired rational thickness and orientation. Due to the symmetry properties of discrete lines, without loss of generality, we limit our study to the first octant where $0 < a < b$ with a and b being mutually prime¹. The plotting algorithm is explained for the naive line $D(a, b, \gamma, b)$ and can be extended to thin and thick discrete lines. It has been first described by Reveillès [13].

¹ Horizontal, vertical and 45° oriented discrete lines can have only thicknesses in integer steps. They are therefore not usable for discrete line juxtaposed halftoning.

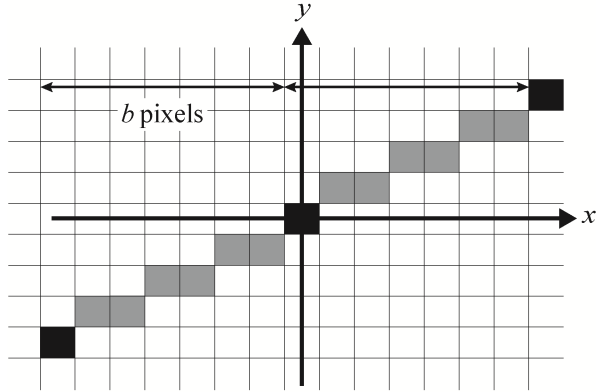


Fig. 2. Illustration of the b -periodicity of discrete lines. A discrete line with a slope equal to a/b repeats the same structure every b pixels; here $a = 4$ and $b = 7$.

A naive digital line is a single-valued function along one of the main axes. With $0 < a < b$, for a given value of x , there is one single value of y such that

$$y = \left[\frac{ax - \gamma}{b} \right] \quad (5)$$

where the square bracket denotes the quotient of the Euclidean division. The value of y selects the pixels whose centers are located on or below the superior continuous line (Eq. (3)) and not located on or below the inferior continuous line (Eq. (4), with $w = b$). In order to derive an incremental formulation for drawing the naive lines, $y(x+1)$ can be written as

$$y(x+1) = \left[\frac{ax - \gamma}{b} + \frac{a}{b} \right] \quad (6)$$

Using the classical identity

$$\left[\frac{\delta + \nu}{\varepsilon} \right] = \left[\frac{\delta}{\varepsilon} \right] + \left[\frac{\nu}{\varepsilon} \right] + \left[\frac{\left\{ \frac{\delta}{\varepsilon} \right\} + \left\{ \frac{\nu}{\varepsilon} \right\}}{\varepsilon} \right] \quad (7)$$

where δ , ν and ε are integer numbers and the curly bracket denotes the Euclidean remainder, we get:

$$y(x+1) = \left[\frac{ax - \gamma}{b} \right] + \left[\frac{a}{b} \right] + \left[\frac{\left\{ \frac{ax - \gamma}{b} \right\} + \left\{ \frac{a}{b} \right\}}{b} \right] \quad (8)$$

Since $b > a$, $[a/b] = 0$ and $\{a/b\} = a/b$, we obtain

$$y(x+1) = y(x) + \left[\frac{r(x) + a}{b} \right] \quad (9)$$

where

$$r(x) = \left\{ \frac{ax - \gamma}{b} \right\} \quad (10)$$

Therefore, an increment in x results in an increment of one unit in y or in no change. The corresponding *remainder function* $r(x)$ is either increased by a or, respectively increased by a and decreased by b . The resulting naive line incremental

plotting algorithm is:

```

y = Div (a * x - γ, b)   Integer division
r = Rem (a * x - γ, b)  Integer remainder
for x = 0 to x = b - 1 do
  Plot Pixel (x, y)
  r = r + a
  if r >= b
    r = r - b
    y = y + 1
  end if
end for
    
```

In a similar manner as Bresenham's Algorithm [16], this algorithm generates incrementally a set of integer coordinates which compose a digital line having a single pixel height [15].

When plotting a thin or thick discrete line of a thickness other than unity, instead of directly plotting the discrete line $D(a, b, \gamma, w)$, we synthesize a top and a bottom naive line with parameters (a, b, γ, b) and (a, b, γ_{new}, b) respectively, such that

$$\gamma_{new} = \gamma + bt \quad (11)$$

where t is the vertical thickness of desired discrete line. The plotted thin or thick discrete line is composed of pixels with pixel centers located between these two naive lines. Pixels belonging to the top naive line and, at the same time, not belonging to the bottom naive line as well as in between pixels are plotted. For a thin line, a pixel may belong to both the top and the bottom naive line. In this case, the pixel is left blank.

IV. BILEVEL SCREEN ELEMENT GENERATION

Classical ordered dither halftone methods rely on dithering with dither matrices. In contrast to dithering methods, we create a library of predefined screen elements [17] obtained by synthesizing discrete lines. Library entries are screen elements corresponding to the different colorant surface coverages. Once the screen element library is created, halftone is performed by traversing the output halftone image scanline by scanline and pixel by pixel and by finding the corresponding location in the input continuous tone image. The color at that location determines an entry within the screen element library. The current output pixel location determines the location within the screen element whose colorant is to be copied into current output pixel. Let us first present the generation of screen elements for black and white halftone and then, in Section V, extend the algorithm to color halftones.

The goal is to generate screen elements made of discrete lines and to halftone an input image by paving the output image plane with these discrete screen elements. The screen element is a discrete parallelogram whose surface is segmented into black and white parts according to the desired black/white surface coverages. These parallelogram screen elements are created using discrete lines of appropriate subpixel thicknesses. The parallelogram forming the screen element is defined by its sides given by vectors $[0 \ T]^T$ and $[b$

$a]^T$ where T is the vertical thickness of the discrete line forming a complete discrete parallelogram and a/b is the discrete line slope. Hence, within the parallelogram screen, a discrete line segment may have a vertical thickness between 0 and T . As an example, Fig. 3 shows a parallelogram screen with 45% surface coverage.

In order to establish the monochrome screen element library, the bilevel screen elements are generated level by level by creating each time within the parallelogram tile a “black” discrete line having a vertical thickness from 0 to T . A discrete line with a thickness of 0 (an empty set of pixels) corresponds to the screen element with 0 surface coverage. A discrete line with vertical thickness T corresponds to the screen element with a surface coverage of 1, i.e. a black parallelogram.

In order to create the halftoned output image, the image plane is paved by replicating the parallelogram screen element along its side vectors $[b \ a]^T$ and $[0 \ T]^T$. Paving the image plane with a discrete parallelogram of the same first side vector $[b \ a]^T$ and a second side vector different from $[0 \ T]^T$, but located within the same discrete line of parameters a and b and of vertical thickness T would yield the same screen. We therefore restrict our attention to the discrete parallelogram given by vectors $[b \ a]^T$ and $[0 \ T]^T$.

Instead of using parallelogram screen elements, we can produce equivalent rectangular screen elements tiling the plane according to Holladay’s algorithm [18]. Given a discrete parallelogram with sides $[p_x \ p_y]^T$ and $[q_x \ q_y]^T$, Holladay’s algorithm yields an equivalent L by H rectangular tile (Fig. 3). Paving the image plane with this rectangular tile is equivalent to paving the plane with the original discrete parallelogram. Parameters of the equivalent rectangular tile are

$$H = \text{GCD}(p_y, q_y); \quad L = \frac{p_x q_y - q_x p_y}{H} \quad (12)$$

Note that the discrete parallelogram surface is $S = p_x q_y - q_x p_y$. The L by H Holladay rectangular tile paves the plane by being replicated horizontally as well as diagonally [18] with replication vector (t_x, t_y) where

$$t_y = H; \quad t_x = \frac{-v \cdot S + H \cdot p_x}{p_y} \quad (13)$$

For our special case with parallelogram side vectors $[0 \ T]^T$ and $[b \ a]^T$ the replication vector (t_x, t_y) is

$$t_y = \text{GCD}(T, a); \quad t_x = v \cdot b \quad (14)$$

In (13) and (14) v is an integer which must be determined such that $0 < t_x \leq L$. Note that all Holladay tile parameters L , H , t_x and t_y are integers.

In order to create an L by H equivalent rectangular tile, the discrete parallelogram screen element is repeated as many times as necessary along the parallelogram vectors until it covers the surface inside the L by H rectangle. The final screen element library size is $L \times H \times (S+1)$, where L and H are derived from Holladay’s algorithm and $S+1$ is the number of possible surface coverage levels (see Section VI).

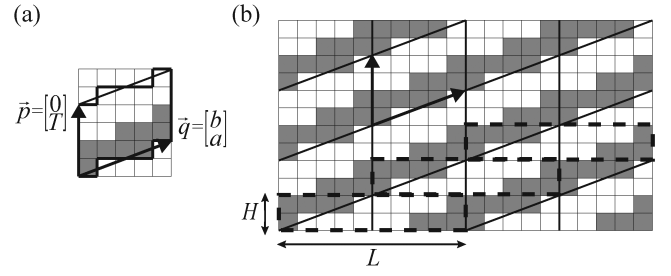


Fig. 3. (a) Parallelogram screen element and its associated vectors and (b) paving a 20×12 output image with that screen element. Surface coverage is 45%. The vertical thickness T is 4 and the slope is $m = 2/5$. The equivalent Holladay tile and its replications are also shown. The discrete line and Holladay’s tile parameters are $a = 2$, $b = 5$, $L = 10$, $H = 2$, and $t_x = 5$.

V. SYNTHESIS OF JUXTAPOSED COLOR SCREENS

Juxtaposed color halftoning relying on discrete lines aims at creating parallelogram screen elements within which successive discrete line segments are associated to different colorants. However, trying to create a screen element library containing the screens for each combination of colorants at every surface coverage level would need a very large memory. For example in case of 8 colorants, with a screen element having 256 distinct surface coverage levels, there are almost 1.55×10^{13} different screen elements (see Appendix).

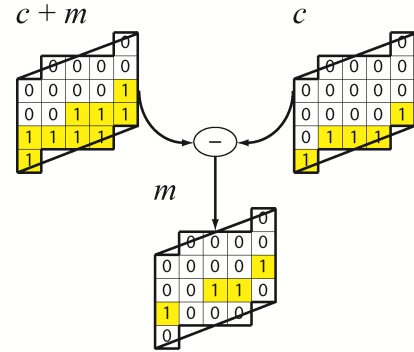


Fig. 4. Example of creation of the magenta part of a screen element by subtracting the tile of surface coverage c from the tile of surface coverage $c + m$.

Instead of precomputing a fully populated color screen element library, we synthesize screen elements for multiple juxtaposed colorants by accessing several times the bilevel screen element library created for a single colorant (Section IV). At halftoning time, we compute the discrete line segments for the second, third and i^{th} colorant by subtracting the screen element corresponding to the sum of the surface coverages of the previous colorants from the screen element representing the sum of the surface coverages of the new and the previous colorants. Note that screen elements are bilevel arrays containing 1 and 0 which, in the present work, represent black and white pixels, respectively. As an example, let us consider a screen of 25% cyan, 20% magenta and 10% yellow surface coverages with the halftoning order cyan, magenta and yellow. The cyan tile is picked directly from the bilevel screen

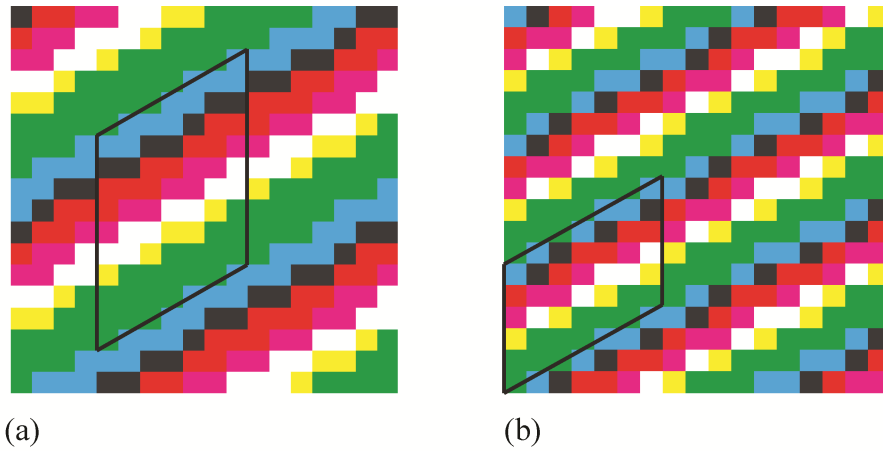


Fig. 5. (a) A juxtaposed halftone screen of orientation $a = 4$, $b = 7$ and vertical thickness $T = 10$ with surface coverages of green: $20/70$, yellow: $5/70$, white: $9/70$, magenta: $8/70$, red: $10/70$, black: $7/70$, blue: $0/70$ and cyan: $11/70$. (b) Another screen with the same parameters but with a smaller vertical thickness $T = 6$. The corresponding parallelogram screen tiles are also shown.

element library by finding the screen tile corresponding to its surface coverage of 25%. The magenta screen is obtained by subtracting the screen tile associated to the surface coverage of cyan (25%) from the screen tile corresponding to the addition of the surface coverages of cyan and magenta, i.e. 45%. This yields the magenta part of the halftone (See Fig. 4). Similarly, subtraction of the screen tile assigned to cyan and magenta (45%) from the screen tile of cyan, magenta and yellow (55%) yields the screen tile for the yellow layer.

More formally, multi-colorant juxtaposed halftoning is performed by calculating the sequence of screen colorant tiles $S_1(c_1), S_2(c_2), \dots, S_K(c_K)$

$$\begin{aligned} S_1(c_1) &= S_B(c_1) \\ S_2(c_2) &= S_B(c_1 + c_2) - S_B(c_1) \\ &\dots \\ S_K(c_K) &= S_B(\sum_{i=1}^K c_i) - S_B(\sum_{i=1}^{K-1} c_i) \end{aligned} \quad (15)$$

where we assume that the current output image location is to be printed with c_1, c_2, \dots, c_K surface coverages of the K colorants and $S_B(c)$ is the screen colorant tile associated to surface coverage c stored in the bilevel screen element library. This approach does not limit the number of colorants. Fig. 5a shows an example of a juxtaposed halftone screen, printed with the cyan, magenta, yellow, blue, green, red, black and white Neugebauer primaries.

VI. SCREEN ELEMENT PROPERTIES, TRADE-OFFS AND SUPERSCREEN

The number of possible surface coverages different from zero is given by the area of the discrete parallelogram tile, i.e. the cross product of the parallelogram vectors, $[0 \ T]^T \times [b \ a]^T = bT$. There is a trade-off between the size of the parallelogram screen tile determining the number of possible surface coverages of the colorants and its visibility. A large screen tile provides many variations of colorant

surface coverages and therefore a precise tone reproduction, but depending on the output device resolution, may become visible by the naked eye.

For a fixed line slope $m = a/b$, let us investigate the role of the vertical thickness T . The number of possible colorant surface coverages increases proportionally with the vertical screen tile thickness T , yielding a larger number of discrete colors and larger screen dots. With a large screen tile thickness T , small surface coverages still yield connected lines. With a smaller thickness T , the same surface coverages result in thin, disconnected lines. Compare Fig. 5a which shows a screen tile with a vertical thickness of $T = 10$ and Fig. 5b which shows a screen tile with $T = 6$ where the same colorant surface coverages are reproduced. In the smaller screen tile of Fig. 5b, the colorant discrete lines are thin and therefore less clustered. This may cause a problem on some printers, where single isolated pixels of a given colorant either have a strong dot gain or tend to disappear.

The main drawback of a large vertical thickness T is the low halftone frequency. According to Fig. 6, the screen frequency is

$$f = \frac{d}{h} = \frac{d}{T \cos \alpha} = \frac{d\sqrt{a^2 + b^2}}{b \cdot T} \quad (16)$$

where f is the screen frequency in screen elements per inch (lpi), d is the printer resolution in pixels per inch (dpi) and α is the angle of the screen element parallelogram support line with $\cos \alpha = b / \sqrt{a^2 + b^2}$. Due to symmetry, it is enough to consider $0 < \alpha < 45^\circ$. Increasing T leads to a decrease of the halftone frequency and therefore to more visible halftones.

The influence of angle $\alpha = \arctan(a/b)$ is another interesting factor. According to (16), angles close to 45° deliver the highest halftone frequencies. This is of high interest since the human visual system has less acuity for diagonal lines than for vertical and horizontal lines. However, an angle of exactly 45° is a special case in which $a = b = 1$ where only integer vertical thicknesses are possible and therefore the number of different colorant surface coverages is limited to T . Nevertheless, angles close

to 45° have a relatively high frequency, offer bT different thicknesses and are less visible by the human visual system.

A large vertical screen tile thickness T and/or large values of b (and consequently of a) result in a large number of colorant surface coverages and a small *surface color discretization error*. The term surface color discretization error describes the difference between the intended surface coverage of a particular colorant and the actual generated surface coverage which is limited by screen element surface, i.e. the number of screen element cells.

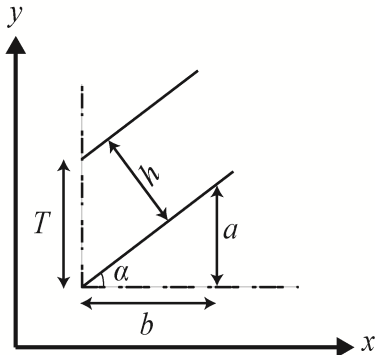


Fig. 6. Relationships between screen parallelogram orientation α , vertical thickness T , and screen period h .

As presented, there is a trade-off between small surface color discretization error due to large values of T and/or b and the visibility of the screen element. In order to keep the surface color discretization error small, we would like to increase the number of surface coverage levels without making the screen element more visible. This can be obtained by extending the current discrete parallelogram screens to superscreens. Superscreens provide more surface coverage levels without modifying the shape of the screen dots.

Classical superscreens may be created by replicating a dither tile a number of times, e.g. 4 times, by multiplying accordingly the dither threshold levels, and by distributing intermediate threshold values among the replicated tiles [12]. Fig. 7a shows an example of a dither tile having originally 5 dither levels, which thanks to the superscreen construction, is expanded to 20 levels.

In order to create a superscreen for discrete line halftoning, we create a large discrete line parallelogram tile containing the desired number of surface coverages as superscreen. We split the parallelogram superscreen vertically into a desired number of sub-tiles, e.g. two sub-tiles. The intended surface coverage of each colorant is then distributed as equally as possible between the sub-tiles of the superscreen. For this purpose, the vertical thickness of the superscreen is split into rational vertical sub-thicknesses of similar size. Note that if the sub-tiles have significantly different areas or if the colorant surface coverages within the sub-tiles are different, they may generate a beating effect visible on smooth color surfaces.

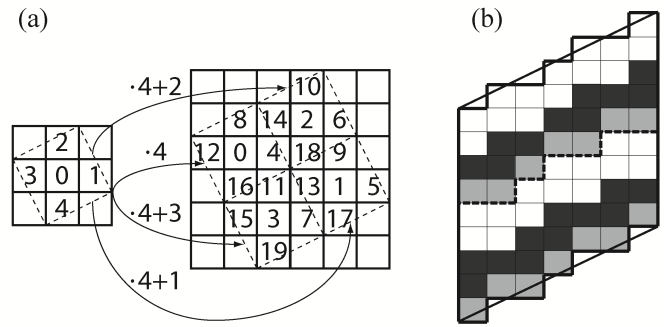


Fig. 7. (a) A classical superscreen where the original tile with 5 dither values is expanded to a superscreen of 20 dither values. (b) A discrete line superscreen composed of two smaller, but slightly different sub-tiles filled with two colorants. Within the sub-tiles, the colorants may have slightly different surface coverages.

Fig. 7b shows a parallelogram superscreen with vertical thickness of $T = 9$ divided into two sub-tiles. The screen frequency appearing to the eye is the screen frequency of the sub-tile, whereas the number of surface coverages is given by the area of the superscreen.

VII. RESULTS

In this section we evaluate juxtaposed discrete line halftoning by a few examples. The selected parameters are $m = 4/7$ and $T = 15$. The vertical thickness T is divided into two rational sub-tile thicknesses of $52/7$ and $53/7$. This leads to a halftone with 106 surface coverage levels having a sub-tile line frequency of 92.14 lpi at a device resolution of 600 dpi. The halftone images are printed with the 8 Neugebauer primaries whose surface coverages are calculated from amounts of cyan, magenta and yellow inks. For the sake of simplicity, color management and gamut mapping are excluded from the present analysis. Therefore, the reproduced colors may differ from the original image colors. The presented examples are printed with classical inks on paper. However, to show the full power of the presented approach, they should be printed with opaque or with metallic inks. We start with an input image with given amounts of cyan (c), magenta (m) and yellow (y) inks and calculate the surface coverages of the eight Neugebauer primaries according to the Demichel equations [19] as follows

$$\begin{aligned}
 a_c &= c(1-m)(1-y) \\
 a_m &= m(1-c)(1-y) \\
 a_y &= y(1-c)(1-m) \\
 a_b &= cm(1-y) \\
 a_g &= cy(1-m) \\
 a_r &= my(1-c) \\
 a_k &= cmy \\
 a_w &= (1-c)(1-m)(1-y)
 \end{aligned} \tag{17}$$

where a_c , a_m , a_y , a_b , a_g , a_r , a_k , and a_w are the surface

coverages of the colorants cyan, magenta, yellow, blue, red, green, black and white, respectively.

Figs. 8, 9 and 10 show the results on different images. As can be seen from these figures, discrete line juxtaposed halftoning enables printing with many different colorants, preserves the edges, reproduces textures at a frequency higher than the screen frequency and does not introduce false contours or other significant artifacts on smooth color gradients. It is therefore well adapted for printing with custom inks and special effect inks.

As in any other juxtaposed halftoning algorithm, the framework assumes high registration accuracy. Misregistration of colorant layers may yield significant color shifts [20], especially when printing with opaque or metallic inks.

VII. CONCLUSION

We presented a new juxtaposed color halftoning method based on discrete lines. It relies on a library of bilevel screen elements created by successions of discrete lines. As many colorants as desired can be juxtaposed within a single screen element. Discrete line halftones provide colorant segments having subpixel precision. The quality of the

resulting halftones is comparable to classical line screen halftones. Juxtaposed halftoning requires however a high registration accuracy. Such a high registration accuracy between colorant layers can be achieved on modern digital printers.

Juxtaposed halftoning can be a solution to many problems. It enables printing with partially or fully opaque inks. Therefore, printing halftone images with custom inks, iridescent inks and metallic inks becomes possible. Application domains comprise flexible packaging, poster design and art. In addition, the surface coverages of all colorants are exactly controlled, i.e. freely chosen relative amounts of cyan, magenta, yellow, blue, green, red and black colorants can be reproduced. This is an advantage in respect to classical rotated clustered-dot screens since independent selection of every colorant surface coverage may yield a larger color gamut and less usage of inks [8], [21].

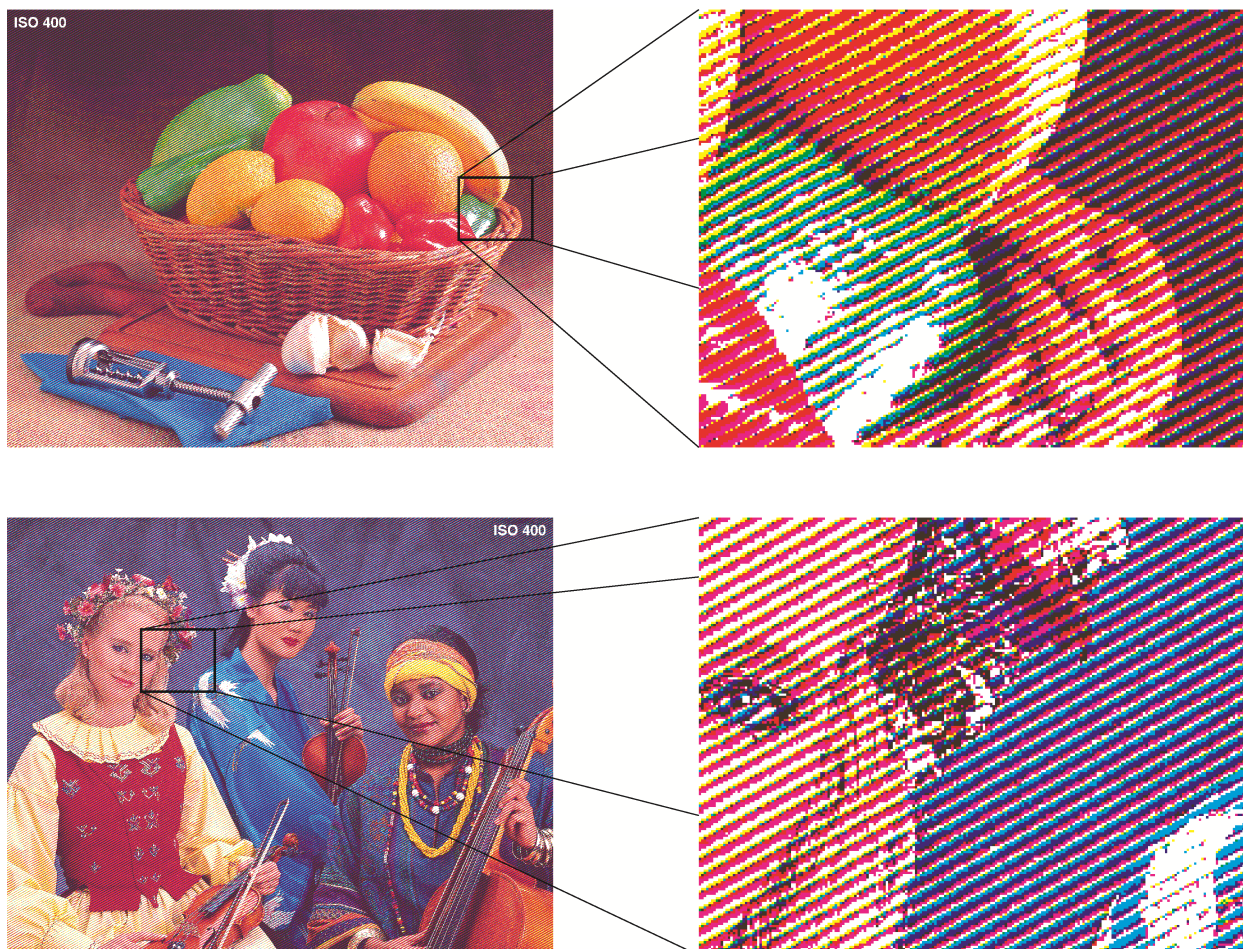


Fig. 8. The “Fruits” and “Orchestra” halftone images reproduced by discrete line juxtaposed halftoning. Discrete line screen element parameters are $a = 4$, $b = 7$ and $T = 15$ with two nearly equal inner rational sub-tile thicknesses. The image is produced at a resolution of 600 dpi, with a sub-tile line frequency of 92 lpi.

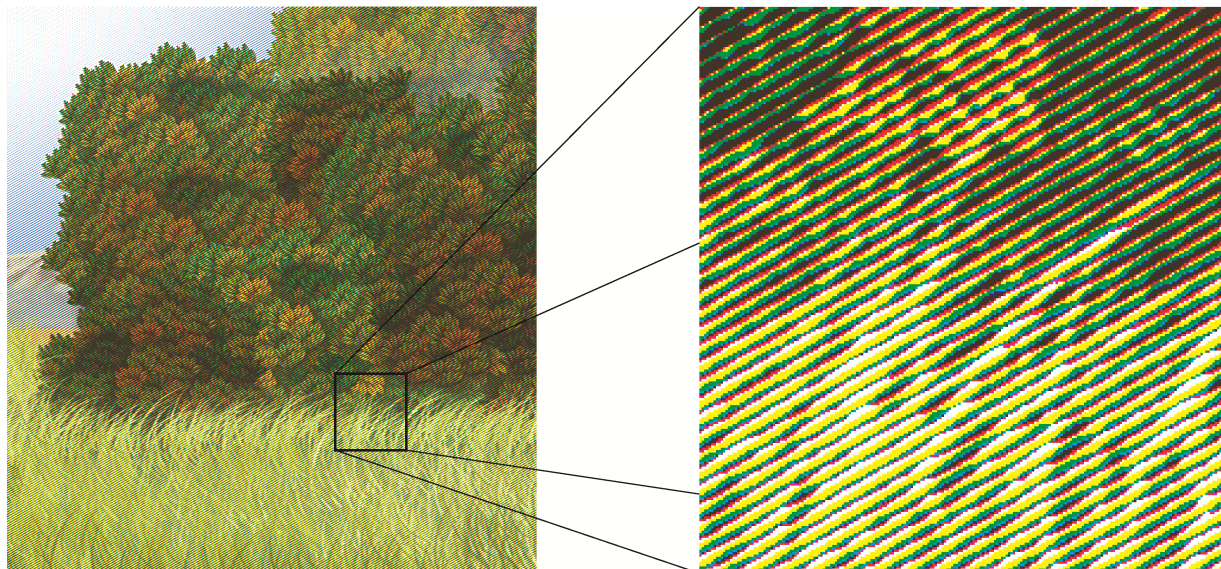


Fig. 9. An example of a highly textured image with the same halftoning parameters as in Fig. 8.

The limitations of the current juxtaposed line halftoning algorithm suggest directions for future work. The proposed juxtaposed line screens rely on parallelograms covering the full length of the screen element. In the future, it would be desirable to create juxtaposed dot screens instead of line screens, i.e. to divide the parallelogram screen surface into parallelogram sub-shapes to be allocated to the different colorants.

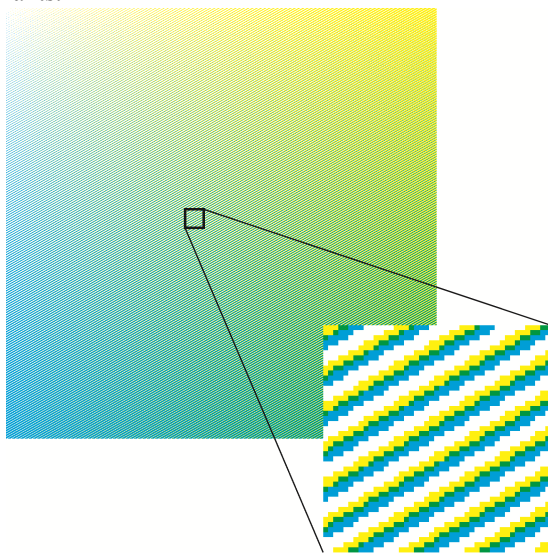


Fig. 10. Cyan-yellow 2D color gradient image with cyan-yellow varying from 0 to 75%, halftoned using surface coverages of the cyan, yellow and green colorants obtained from the Demichel equations.

APPENDIX

To compute the number of combinations of K colorant values, including paper white, such that each colorant can take an integer from 0 to N and the addition of these values is equal to N , one should solve the following equality

$$x_1 + x_2 + \dots + x_K = N \quad (18)$$

where paper white is considered as the K^{th} colorant. The total number of solutions is the number of non-negative solutions of (18) which is known as the number of ways to distribute N indistinguishable balls into K distinguishable boxes [22]

$$\binom{N + K - 1}{N} \quad (19)$$

For example if we consider a screen with 8 colorants and each colorant takes the value between 0 and 255, there are

$$\binom{255 + 8 - 1}{255} \approx 1.55 \times 10^{13} \text{ possible combinations.}$$

REFERENCES

- [1] F. A. Baqai, J. H. Lee, A. U. Agar, and J. P. Allebach, "Digital color halftoning," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 87–96, 2005.
- [2] H. Kueppers, "Printing process where each incremental area is divided into a chromatic area and an achromatic area and wherein the achromatic areas are printed in black and white and the chromatic areas are printed in color sub-sections," U.S. Patent 4 812 899, March 14, 1989.
- [3] R. V. Klassen, R. Eschbach, and K. Bharat, "Vector error diffusion in a distorted colour space," *Proc. IS&T 47th Annual Conf.*, 1994, pp. 489–491. Reprinted in *Recent Progress in Digital Halftoning*, R. Eschbach Ed. *IS&T Publ.* 1994.
- [4] Z. Fan and S. Harrington, "Improved quantization methods in color error diffusion," *J. Electron. Imaging*, vol. 8, no. 4, pp. 430–438, 1999.
- [5] Z. He, "Hierarchical error Diffusion," *IEEE Trans. Imag. Process.*, vol. 18, no. 7, pp. 1524–1535, 2009.
- [6] V. Ostromoukhov and R. D. Hersch, "Multi-color and artistic dithering," *Proc. SIGGRAPH 99, in Computer Graphics Proceedings, Annual Conference Series*, pp. 425–432, 1999.
- [7] R. D. Hersch, P. Donz , and S. Chosson, "Color images visible under UV light," *ACM. Trans. Graphics (Proc. ACM SIGGRAPH)*, vol. 22, no. 3, pp. 427–436, 2007.
- [8] J. Morovi , P. Morovi , and J. Arnabat, "HANS – A new color separation and halftoning paradigm," in *Proc. IS&T 18th Color Imaging Conference*, 2010, pp. 359–364.

- [9] J. Morović, P. Morović, and J. Arnabat, "HANS: Controlling ink-jet print attributes via Neugebauer primary area coverages," *IEEE Trans. Imag. Process.*, vol. 21, no. 2, pp. 688–696, 2012.
- [10] L. Velho and J. Gomes, "Digital halftoning with space filling curves," *Proc. SIGGRAPH 91*, pp. 81–90, 1991.
- [11] S. Chosson and R. D. Hersch, "Visually-based color space tetrahedrizations for printing with custom inks," in *Proc. Color Imaging Conference: Device-Independent Color, Color Hardcopy and Graphic Arts VI, SPIE* vol. 4300, 2001, pp. 81–92.
- [12] C. Hains, S. G. Wang, and K. Knox, "Digital color halftones," in *Digital Color Imaging Handbook*, G. Sharma, Ed. CRC Press, 2002, pp. 385–490.
- [13] J. P. Reveillès, "Géométrie discrète, calcul en nombres entiers et algorithmique," Ph.D. dissertation, University of Louis Pasteur, Strasbourg, 1991.
- [14] J. P. Reveillès, "Combinatorial pieces in digital lines and planes," *SPIE Vision Geometry IV*, 2573 1995, pp. 23–34.
- [15] R. Klette and A. Rosenfeld, *Digital Geometry*. San Francisco: Elsevier, 2004, ch. 9.
- [16] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.
- [17] P. Li and J. P. Allebach, "Look-up-table based halftoning," *IEEE Trans. Imag. Process.*, vol. 9, no. 9, pp. 1593–1603, 2000.
- [18] T. M. Holladay, "An optimum algorithm for halftone generation for displays and hard copies," in *Proc. SID*, vol. 21, 1980, pp. 185–192.
- [19] D. R. Wyble and R. S. Berns, "A critical review of spectral models, applied to binary color printing," *Color Research and Application*, vol. 25, no. 1, pp. 4–19, 2000, with the Demichel equations originally published in M. E. Demichel, "Procédé," vol. 26, pp. 17–21, 1924.
- [20] B. Oztan, G. Sharma, and R. P. Loce, "Misregistration sensitivity in clustered-dot color halftones," *J. Electron. Imag.* vol. 17, no. 2, pp. 023004, 1–30, 2008.
- [21] W. L. Rhode, and C. M. Hains, "The influence of halftone orientation on color gamut and registration sensitivity," in *Proc. IS&T 46th Annual Conference*, 1993, Cambridge, MA, Reprinted in *Recent Progress in Digital Halftoning*, R. Eschbach Ed. *IS&T Publ.* 1994, pp. 117–119.
- [22] K. H. Rosen, *Discrete Mathematics and Its Applications*. 6th Ed. New York: McGraw-Hill, 2007, pp. 370–382.



Vahid Babaei obtained his M.S in Color Science from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, 2010.

He is currently a PhD student at School of Computer and Communication Sciences of Ecole Polytechnique Fédérale de Lausanne (EPFL), where he works with the group of Prof. R. D. Hersch. His research focuses on color reproduction and color halftoning.



Roger D. Hersch received the engineering and PhD degrees from ETH Zurich in 1975 and from EPFL in 1985, respectively. He is a professor of computer science and head of the Peripheral Systems Laboratory at the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

He has published more than 150 scientific papers, is the editor of several books, and is inventor or co-inventor in many patent applications. He is interested in novel imaging

techniques related to color prediction, color reproduction, artistic imaging, and security printing. He is a member of the IEEE Computer Society, and a fellow of IS&T (Society for Imaging Science and Technology).