Direct Pattern Control Halftoning of Neugebauer Primaries

Peter Morovič, Ján Morovič, Jay Gondek, and Robert Ulichney, *Senior Member, IEEE*

***Abstract*— Halftoning is a key stage of any printing image processing pipeline. With colorant-channel approaches, a key challenge for matrix-based halftoning is the co-optimization of the matrices used for individual colorants, which becomes increasingly complex and over-constrained as the number of colorants increases. Both choices of screen angles (in clustered-dot cases) or structures, and control over how individual matrices relate to each other and result in over-versus side-by-side printing of the colorants, impose challenging restrictions. The solution presented in this paper relies on the benefits of a halftone area Neugebauer separation pipeline, where local Neugebauer Primary use is specified at each pixel and where halftoning can be performed using a single matrix, regardless of the number of colorants. The provably complete plane dependence of the resulting halftones will be presented among the solution's benefits.**

***Index Terms*— Digital printing, image processing.**

I. Introduction

THE color and image processing pipeline of any printing system needs to address at least the following questions: how does color content get adjusted to the capabilities of a given printing system (i.e., color management), how are the system's colorants combined to match the colors of its gamut (i.e., color separation) and how color separation choices are translated into discrete colorant amount placement in the final print (i.e., halftoning).

Traditionally, color separation consisted in answering the question of how much of each available colorant to use for matching each color within a printing system's color gamut and halftoning was responsible for making spatial choices of where to apply each colorant in turn, given the choices of colorant amounts made during color separation. Halftoning colorant amounts one by one can lead to unwanted interactions between their halftone patterns. To mitigate this, previous work focused on avoiding unwanted interactions. For the case of clustered-dot halftones, individual threshold matrices are designed to minimize moiré effects (e.g., [1]). For dispersed-dot halftoning, local neighborhood operations are needed to provide "plane-dependency" between specific colorants, allowing for their placement to be kept apart for where possible; this was performed for error diffusion (e.g., [2], [3]), along with a parallelized version [4]. Related to the approach presented in later in this paper is a class of stacked multi-toning error diffusion algorithms [21], [22]. Here, in the simpler case of a single colorant, multiple levels are error-diffused such that the resulting combined pattern involving multiple levels of a colorant has the desired properties (e.g., blue-noise). This involves a sequential process applied to an area-coverage ordered hierarchy of tones, where the tone with highest coverage is error diffused first and subsequent tones of decreasing coverage are error-diffused over the locations used by higher-coverage levels. The approach also extends to multiple colorants and their combinations, processed in order of decreasing area coverage and to multi-scale processing [22]. While such approaches offer some control over colorant combinations, they are constrained by operating in a specific order of decreasing coverage and by diffusing error between at-pixel colorant combinations.

Many of the above limitations are a consequence of acting on colorant amounts determined by color separation. When the domain in which color separation specifies printed output changes, so do the constraints and opportunities for halftoning. This is what the introduction of HANS [5] brings to the table, where color separation no longer specifies colorant amounts, but where it determines what relative area coverage to assign to each of a system's Neugebauer Primaries (NPs). Hence, for a binary printing system with three inks, where color separation previously specified continuous amounts of CMY, a HANS pipeline specifies relative area coverages for eight NPs: blank substrate, C, M, Y, CM, CY, MY, CMY. Instead of the result of color separation being instructions that apply colorant-by-colorant, they become more akin to selecting numbers of tiles of different colors with which to fill a given area of a mosaic (Fig. 1).

More specifically, the output of color separation is in the form of NP area coverages (NPacs) per pixel, such as the following example taken from our simple CMY system: [w,C,MY] = [0.6,0.3,0.1], which specifies that 60% of some local area be left blank, 30% be covered by cyan (C) and 10% contain the combination of magenta and yellow (MY). Table 1 summarizes the main differences between a colorant-based domain and a HANS domain.

The differences shown in Table 1 due to a change in domain from colorants to NPacs mean that prior art on halftoning algorithms does not directly apply and new ways to map

Manuscript received July 4, 2016; revised December 4, 2016, March 20, 2017, and May 26, 2017; accepted May 26, 2017. Date of publication June 8, 2017; date of current version July 6, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo. *(Corresponding author: Peter Morovič.)*

P. Morovič and J. Morovič are with HP Inc., 08174 Sant Cugat del Valles, Spain (e-mail: peter.morovic@hp.com; jan.morovic@hp.com).

J. Gondek is with HP Inc., Vancouver, WA 98683 USA (e-mail: jay.s.gondek@hp.com).

R. Ulichney is with HP Labs, Stow, MA 01775 USA (e-mail: u@hp.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2713939

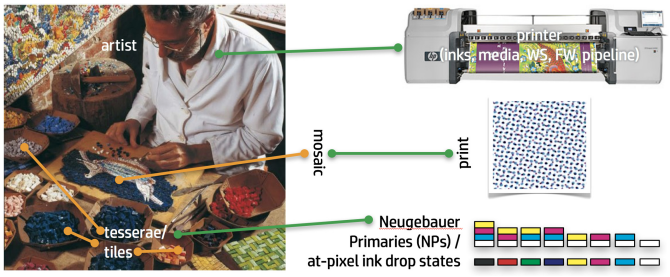

Fig. 1. HANS print control as mosaic building.

TABLE I

OVERVIEW OF A COLORANT BASED APPROACH TO PRINT CONTROL AND HANS

	Colorant based	HANS
Domain	Colorant (ink) channels	Neugebauer Primary area coverages
What is controlled	Amount (%) of each colorant	Area coverage (probability) of each at-pixel state (NP)
Dimensionality	# of colorants (e.g. 4 for CMYK)	# of Neugebauer Primaries (e.g. 16 for binary CMYK)
Role of Halftoning	Spatial distribution per colorant channel –determines overprinting statistics jointly with color separation	Spatial distribution of at-pixel halftone values (NPs) – explicit control over overprinting is exercised by color separation alone

from NPacs to NPs (the HANS equivalent of mapping from colorant %-ages to NPs, which is the role of colorant channel halftoning) need to be developed. Colorant based matrix halftoning applies halftone matrices to each of the colorant channels individually (with some level of co-optimization being possible) and the resulting halftone is the superposition of colorant channel halftones and the respective choices made at each colorant level. This cannot be done in NPac terms. While it is possible to think of NPs as the HANS equivalent of colorant channels, this analogy has limitations since NPs are strictly complementary. Hence, performing the equivalent of colorant channel based matrix halftoning for a Cyan (C) NP plane and an NP plane that represents the overprint of C and Magenta (M), must result in a halftone that never places C at a location where C and M have been placed and vice-versa. This is not an issue in colorant based halftoning where a C plane and a M plane are halftoned separately and their joint, superposed halftones determine their coincidence (overprint).

Likewise, in colorant based error diffusion algorithms, what is halftoned are colorant channels and the error is incurred in colorant proportions. Again, this does not apply to the HANS domain directly, whether in its simplest form, or advanced evolutions of error diffusion methods, such as those using Direct Binary Search [16], hierarchical [17], [18] or multiscale [19] approaches, since they still operate in colorant channels. A key point to note here is that while for a given NPac there is a

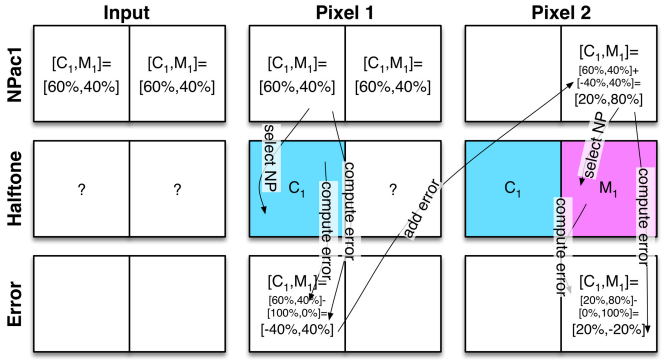

Fig. 2. HANS error diffusion example.

unique, direct conversion to a colorant domain, this does not hold vice versa. For example, 50% of C and 50% of M inks (in colorant terms) can in HANS terms mean 50% of C and 50% of M (in at-pixel halftone state, NP terms – meaning half the area is occupied by halftone values of C and half the area is occupied by halftone values of M) or it can mean 50% of C and M overprinted with 50% left blank (again, in at-pixel halftone state, NP terms) or anything in-between. In other words, what in colorant terms is a single colorant vector, in HANS terms is a continuum.

In previous work, the halftoning applied to such NPacs has been error-diffusion based [5], where at a given halftone pixel, one of the NPac's NPs was selected. E.g., in the above example, it could be the blank NP that would be selected for one pixel, resulting in an error of 40% too much blank coverage and C and M plus Yellow (Y) being short by 30% and 10% respectively (Fig. 2). This error would then be diffused to neighboring pixels' NPacs using existing error diffusion [6].

While error diffusion is an excellent way for taking NPacs and constructing colorant patterns from them, it does have several limitations. The first is throughput, since a sequential traversal of halftone pixels is required and even though there have been some ingenious attempts at parallelization [4], even their speeds are well below those of matrix based halftoning, where each halftone pixel can be processed independently. This in turn lends itself to massive parallelization and provides a scalable throughput architecture.

The second challenge of some error diffusion algorithms is that they involve an element of randomness and therefore results in varying halftone patterns for the same input. This results both in challenges for color modeling and for post-processing that require deterministic patterns (e.g., the stitching together of halftones for separate parts of a printing system, or the application of data embedding).

Both these challenges point to the use of matrix-based halftoning for HANS. However, existing solutions, with potentially a different matrix for each colorant and with halftoning done mostly independently per colorant, are not directly applicable to HANS. The key reason for this is the difference in the two domains – the colorant one and the NP one. While the former under-determines halftone patterns, in that multiple patterns of colorant placement can match a given combination of colorant amounts, the latter specifies unique colorant

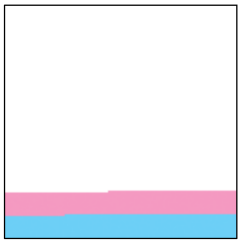

Fig. 3. Halftone of NPac of 80% blank media, 10% of one drop of M and 10% of C over 128 × 128 pixels.

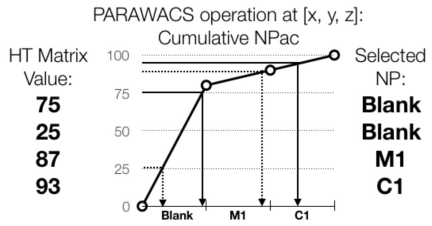

Fig. 4. Uniform random numbers halftoning NPac in Fig. 1.

combination statistics. Hence, the separate use of halftoning matrices is not an option, since their application to on an NP-by-NP basis would introduce even greater constraints on their generation than colorant channels do and since the number of NPs can exceed several hundred or even thousand, this may not be possible.

To address these challenges, the PARAWACS (Parallel Random Weighted Area Coverage Selection) method will be introduced next, followed by a proof of its full plane-dependence and finally an analysis of its pattern quality.

II. PARAWACS

Like in the case of an ink-channel separation, HANS too uses relative proportions of addressable channels, however unlike the ink-channel case, these addressable channels are Neugebauer Primaries and the amounts relate to proportions of device states. For an input, such as RGB device color, a HANS separation uses recipes of NPs, or NP area coverages as its domain. These area coverages can be thought of as implicitly referring to a unit area and the proportions of an NPac therefore express relative sub-areas that need to be occupied by each NP present in the NPac. Hence, one way to think about the role of halftoning here is that for a sufficiently large area of a constant NPac, halftoning should result in a placement of individual NPs such that when counting their frequency over the area, the original area coverage is obtained. This is no different from traditional, ink-channel based separations where ink coverages are specified and halftoning distributes them. There too an area of constant ink-channel coverages, once halftoned, will result in the specified amounts of inks.

So, given an area of 128 × 128 pixels that has a constant NPac of 80% blank substrate, 10% one drop of M and 10% of C, a simple area-coverage-preserving halftoning is sequential placement (Fig. 3). Counting the number of pixels then yields 1638 pixels (10%) of each of M and C and the remaining pixels are left blank (80%). While Fig. 3 clearly is not a desirable halftone, it satisfies the constraint of having distributed the relative area coverages of the NPac.

Another way to think about the example above is the following: given a (sufficiently large) area of a constant NPac, its halftone should have the following property: *The likelihood of picking one of the NPs from the halftone of the area is equal to the area coverage of the input NPac.*

In other words, if we uniformly randomly sampled locations of the halftone in Fig. 3 we would have an 80% chance of picking a blank location, a 10% chance of picking a one drop of cyan ink pixel etc. Nonetheless, Fig. 3 lacks another important attribute of halftoning, which is a uniform spatial distribution of the NPs. Instead, it simply clustered and ordered all states sequentially.

Since we described an NPac as a probability distribution, a simple way to have a more uniform spatial distribution is to generate uniformly distributed random numbers, using a standard random number generator and scaling them to a range of 0 to 100 (the range of the NPacs). Then, depending on the randomly generated value, a different NP is chosen from the NPac, proportionally to its probability or area coverage. To simplify this selection, the NPac can be expressed cumulatively such that the NPac: [Blank 80%, 1 drop M 10%, 1 drop C 10%] or a shorthand (with W denoting blank substrate, typically White (W)): [W = 80%, M = 10%, C = 10%] becomes in cumulative terms [W = 80%, M = 90%, C = 100%], which in turn defines intervals for each of the NPs such that [0 to 80] corresponds to the Blank state, [80 to 90] to one drop of M and [90 to 100] to one drop of C. Given this representation, the random numbers generated simply need to be categorized according to the intervals. If a random number at pixel [x, y] is in the range [0 to 80] it is left blank, if it is in the range [80 to 90] a drop of M NP is placed and if the random number is in the interval [90 to 100] a C drop NP is placed. The diagram in Fig. 4 shows this process for four random values (corresponding to four [x, y] locations):

Below, pseudo-code is shown that corresponds to the process in Fig. 4, performing the halftoning operation at a single pixel. The inputs are a halftone value (i.e., from a halftone matrix generated using methods such as those in [8], [9]), a sequence of NPs arranged in a chosen order (of length n) and a corresponding sequence of area coverages (ACs, of the same length as NPs, obtained using methods such as presented in [5]):

```
Inputs: HT_value, NPs, ACs
% build cumulative probability distribution of ACs
ACs_cumulative = zeros(n, 1);
ACs_cumulative(1) = ACs(1);
for i = 2:n
   ACs_cumulative(i) = ACs_cumulative(i-1) + ACs(i);
end
% find NP index based on HT_value
for i=1:n
   if ACs_cumulative(i) >= HT_value
      NP_selected = NPs(i);
   end
 end
Output: NP_selected
```

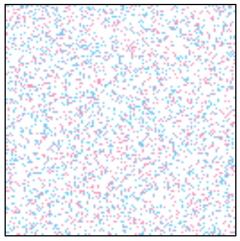

Fig. 5. Uniform random numbers used to halftone the same NPac as shown in Fig. 3.

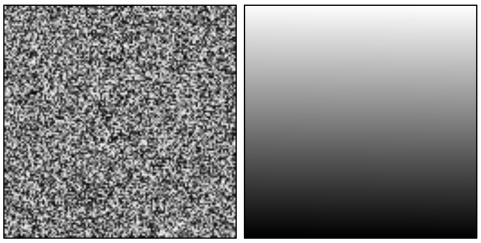

Fig. 6. A grayscale visualization of the uniform random numbers used to produce the halftone in Fig. 5 (left) and of a sequential choice used for the halftone in Fig. 3 (right).

Fig. 5 shows the result of applying this process based on a uniform random number generator. The resulting halftone still satisfies the condition of placing the correct number of each of the constituent NPs of the NPac over the patch, however, it distributes the states more evenly than the naïve, sequential placement in Fig. 3. In halftoning literature, the type of pattern shown in Fig. 5 is often referred to as white noise [7], due to the uniform random nature of the placement.

While these random numbers can be generated on-line, independently for each [x,y] location, doing so means there is no consideration for the overall pattern. And, while Fig. 5 is significantly better than Fig. 3 in terms of evenness, clearly the pattern is noisy and would result in a grainy print. This can further be seen in Fig. 6 (left) which shows all halftone values (the random values used as selectors in the above process) over the 128×128 pixel area.

Note that the halftone in Fig. 3 can also be thought of as corresponding to a matrix of halftone values and Fig. 6 (right) shows this matrix in grayscale. The values in the two examples of Fig. 6 are the same 128×128 grayscale values, except they are placed differently – in one case uniformly randomly and in the other sorted by grayscale value and placed sequentially.

An important property of PARAWACS can be seen here already: the spatial nature of the values used in the selection of NPs is directly translated in the halftone. Hence Fig. 6 can be thought of as *halftone matrices* that can be applied to any NPac or any content. There is a rich body of literature in the field of designing such halftone matrices, even if in a different context. For the case of dispersed-dot halftoning, an example of blue-noise matrix generation is the void-and cluster algorithm [8]; dispersed-dot matrices have also been generated using direct binary search, or DBS [9]. An example of clustered-dot halftoning where clusters are built on blue noise centers is referred to as green noise [7].

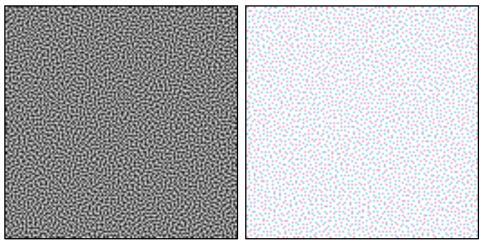

Fig. 7. Blue noise matrix (left) and resulting halftone (right).

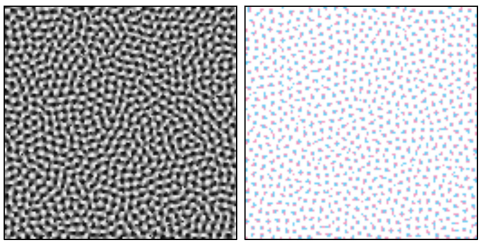

Fig. 8. Green noise matrix (left) and halftone (right).

Figs. 7 and 8 show blue and green noise arrangements – both the halftone matrices as and the result, when applied to the same NPac patch as in Figs. 3 and 5.

Recall that all the above halftones share the property of having the same number of pixels left blank (80%), the same number of pixels occupied by one drop of M (10%) and the same number of pixels occupied by one drop of C (10%), but they differ in the spatial arrangement of the halftone values (the halftone matrix). Hence the above halftone matrices will have a uniform probability of each of the gray-scale values: e.g., using an encoding of [0 to 255] for grayscale values there will be (128*128)/255 pixels of each of the 255 values in the matrix.

Both the precision at which area coverages are encoded (i.e. the % values of the NPac), the precision at which the values of the halftone matrix are generated and the patch size over which an NPac is constant, determine the accuracy with which an input NPac can be reproduced. Again, this is analogous to the colorant channel approach where the accuracy of reproducing colorant proportions depends on similar factors.

Another parameter is the order of NPs in an NPac, based on which the cumulative distribution is built. Whether blank substrate is first or last determines which ranges of values of the halftone matrix get used for its pattern and while in a white-noise matrix this may not matter much, since all values are equally white-noise, once a specifically designed halftone matrix is used, the difference can be significant. A simple example is shown in Figure 9, where the same green-noise matrix is used as in Fig. 8, except reordering the NPs in reverse order (one drop of C first, then one drop of M and finally blank substrate). Notice the difference between Fig. 8 (right) and Fig. 9 where the blank media in one case is the surround of the clusters, while in the other case it is at the center of the clusters.

Another important property of this approach of halftoning, which acts by means of selecting NPs from a predetermined list, and selects them according to a predetermined probability,

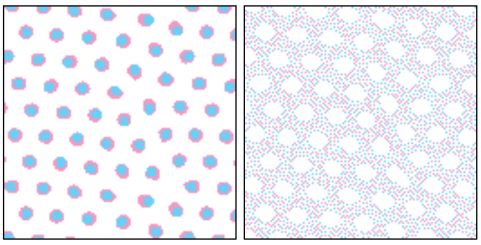

Fig. 9. Halftones an NPac, using one clustered-dot halftone matrix with two NP-orderings: W, M, C (left) & C, M, W (right).

Fig. 10. An image halftoned using a white noise (left), blue noise (center) and green noise (right) matrix.

is that, by definition, no NP that is not in the NPac can be placed in the halftone. While this may seem obvious, it certainly is not the case in colorant-channel halftoning, where the colorant-recipe only determines colorant quantities and it is left up to halftoning to determine how to place each drop of each of the inks individually without full control over whether to place drops on top of each other (dot-on-dot printing or overprinting) or whether to place them next to each other (dot-off-dot printing or side-by-side printing). What this means is that secondaries, such as one drop of C and one drop of M placed at the same [x, y] location, are an indirect consequence of halftoning. This is both true for traditional matrix-based halftoning and error-diffusion, since both act in colorant-channels. While it is possible to correlate (or de-correlate) colorant channels and therefore favor overprinting (or side-by-side printing), the level of control is indirect (overprinting would occur after a certain amount of drops has been placed) and global, meaning that it is not possible to have a pipeline that for some colors uses exclusively overprinting NPs, while for other colors uses exclusively a combination of two single-ink NPs (i.e. side-by-side printing). In a HANS domain and with PARAWACS halftoning, instead, access to all possible NPs and all their possible probability distributions (NPacs) is provided concurrently in a single pipeline and is guaranteed to be maintained in the halftone.

A consequence of the above property is the potential for better color accuracy both at single NPac level since halftoning is deterministic and predictable, as well as better behavior between nodes (combinations of NPacs) since transitioning in the HANS domain remains closed in terms of the constituent NPacs: an NPac that is half-way between two NPacs will have 50% probabilities from each of the two NPacs hence only contain NPs that are present in the two end-point NPacs.

While in the examples used to illustrate the halftoning mechanism, a square patch has been used, this probabilistic understanding of NPacs and their halftoning applies at a single pixel too. As is the case in general image content, every pixel of an input image can vary in terms of RGBs and therefore once a HANS color separation is applied, can have a different NPac. Hence, NPacs are better thought of as probabilities than recipes since at any one pixel a single NP needs to be selected from among the NPs of the NPac. Having image content expressed as an array of such probability distributions at each pixel, converting the array from NPacs to NPs – i.e. halftoning – is by definition a problem of sampling probability distributions, just the same as in the case of the constant NPac over some area. Note again that this is analogous to traditional, ink-channel based domain. Here, the colorant-channel recipes at each pixel are probability distributions of the respective inks too, except in a domain that is not directly related to NPs but where halftoning determines states in each ink-channel and it's their union that indirectly determines the NP to be placed at a location.

To illustrate the halftoning algorithm on more complex content an image is shown in Fig. 10 halftoned using three different halftone matrices described earlier (white, blue and green noise) and a constant NP-ordering (light-to-dark) that has empirically been found to be pleasing for image content.

Another important aspect of the present algorithm is its computational complexity compared to other approaches. Error diffusion for example is notoriously complex and hard to parallelize (some attempts have been made [4]) as well as non-deterministic in that the same colorant-vector can yield different at-halftone states, can depend on content that may not be visually relevant (e.g. a large error carried over many pixels) and a given pixel may end up with a halftone value that does not pertain to its colorant vector (again due to the error vector, the error may accumulate and a halftone pixel may be placed using colorants not present at that pixel). It is also hard to parallelize due to its sequential nature. Colorant based matrix halftoning instead requires as many thresholding steps as there are colorants. PARAWACS instead involves a single operation per halftone pixel that amounts to the comparison of the value of the halftone matrix at that location with the cumulative probability distribution, resulting in the final halftone value directly. This both lends itself well to parallelization, since choices are made independently, and results in a deterministic process: for the same halftone matrix value and the same NPac, the same NP (at-pixel halftone state) is always selected. This means that, at a pixel, only NPs present in its NPac are placed.

Since halftoning is performed in a spatially independent way, PARAWACS also lends itself well to functional halftoning. Here the halftone pattern is not determined by a matrix, but by a spatial function that can be computed at [x, y] given a pre-determined $f\ (x, y)$. Such functions have as their domain the 2D plane, while their image is a discrete interval of values, e.g., from 0 to 255 for an 8-bit encoding. While in matrix design one of the conditions of a halftoning matrix is that of tileability to avoid artifacts like horizontal/vertical edges where the matrices are tiled, in functional halftoning this constraint translates to the function being defined over arbitrary [x, y] values (hence no tiling occurs). Here an

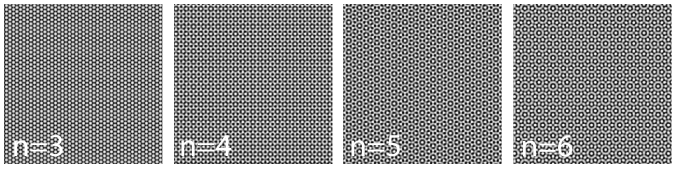

Fig. 11. Examples of quasicrystal patterns for n = 3 to 6.

interesting class of functions are quasicrystals [10] or quasi-periodic crystals, which are aperiodic structured patterns that never line up perfectly with a shifted copy of themselves – they lack translational symmetry. They are ordered but not repeating and while there is no single functional definition for them, the most common approach is to generate them by super-imposing a series of wave functions, which can be computed at any location [x, y] analytically. They form an infinite class of functions that correspond to an infinite set of patterns depending both on spatial location, a constant-offset and the number of waves used in their generation. All these parameters (location, constant offset, number of waves) can be varied and result in different patterns.

Let n denote the number of waves to be added up, x and y refer to spatial location in a plane and t be an offset parameter. Then the value of a quasicrystal function $Q_{n,t}(x, y)$ can be written as follows:

$$Q_{n,t}(x, y) = \sum_i cos(sin(\theta_i)^* x + cos(\theta_i^* y + t))$$

where i is from *1* to n and θ_i is defined as follows:

$$\theta_i = 0 : \pi/n : \pi(1 - 1/n)$$

For n = 6 and a computing the values of $Q_{n,t}(x, y)$ over a grid of 512 × 512 [x, y] locations, the resulting pattern is shown in Fig. 11 (top right), where patterns for n equal to 3, 4 and 5 are also shown.

The above function $Q_{n,t}(x, y)$ will compute values between [0, 1] at any location in the [x, y] plane, however their distribution is not uniform. Since a halftone matrix is expected to have the same frequency for each level, an extra step is required that maps the [0, 1] range with its respective frequency to the appropriate range, say, [0 255]. This is achieved by computing quantiles with a resolution in function of the desired bit-depth of the halftone matrix, pre-computed over some representative area of the function results. Once quantiles are computed, the respective ranges can be mapped to halftone level values, ensuring a uniform distribution.

While this approach does not result in a halftone matrix, its use is the same since instead of comparing the cumulative area coverage distribution against a value from a matrix whereby [x, y] in the image being halftoned and [x, y] in the halftone matrix are aligned, in this case the halftone value can be computed on-line at halftone time, followed by the image-value adjustment (a simple look-up) to achieve a uniform distribution. Note that for this approach to be used at print-time, computational complexity needs to be considered and depending on the complexity of the pattern generating functions and the available processing power this method may or may not be feasible. An alternative to at-print-time computation of values is to pre-compute some area of values ahead of printing sufficient to keep a buffer of values available at print-time. Also, note that the intention here is to simply showcase an alternative to either error diffusion or a pre-computed halftone matrix (such as those used in the rest of this article) that is enabled by PARAWACS and its use of a single value at a given pixel.

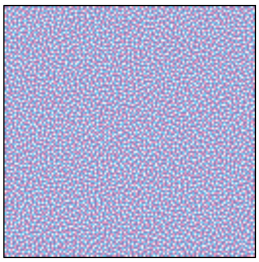

Fig. 12. Halftoned NPac created from Demichel probabilities.

The above types of matrices used with PARAWACS (white noise, blue noise, green noise and the functional quasicrystal patterns) will impact dot-gain as well, as is the case with any other halftoning algorithm. This is a consequence of the inherent patterns rather than the algorithm as such. A clustered-dot placement (such as using an AM screen in traditional halftoning, or a green-noise matrix shown above) will have different (lower) dot-gain than a dispersed-dot (FM screen or blue noise) pattern due to the sizes of the respective areas (clusters) of each of the Neugebauer Primaries.

While so far all examples used the HANS-native NPac domain, PARAWACS can also be used with colorant channels. In this case an ink vector input (a contone specification of amounts of C, M, Y, and K colorants) is first converted to an NPac that is halftoned by PARAWACS. E.g., a document to be printed may be specified in amounts of C, M, Y, and K, thereby precluding a colorant separation step. In this case, the CMYK amounts need to be converted to NPac probability distributions. The task of creating Neugebauer Primary probabilities from uncorrelated CMYK primary probabilities involves determining overlapped printer primaries (CM, CY, MY, CK, MK, ...) and their associated probabilities. Automated creation of NPacs from CMYK can be achieved using a variety of methods. One method is to directly compute NPs using the Demichel equations. This method predicts the NPs based on the probability of uncorrelated overlap [11]. For example, a CMYK of (60%, 60%, 0%, 0%) becomes [C = 24%, M = 24% CM = 36%, W = 16%]. An example of this PARAWACS halftoned NPac is shown in Fig. 12. In contrast to Demichel probabilities, it is often desirable to minimize the probabilities of NPs with overlap of printer primaries, which in turn maximizes overall spatial frequency and thereby reduces visible grain. One method to achieve this is to "stack" K, C, M, Y probabilities and produce overlapping NPs only if the sum of K, C, M, and Y exceeds 100%, like the C, M correlation algorithm described in [12]. Continuing the example, a CMYK of (60%, 60%, 0%, 0%) becomes [C = 40%, M = 40% CM = 20%, W = 0%]. We can generalize this approach by successively creating overlapping NPs until the sum of probabilities (excluding white) is <= 100%.

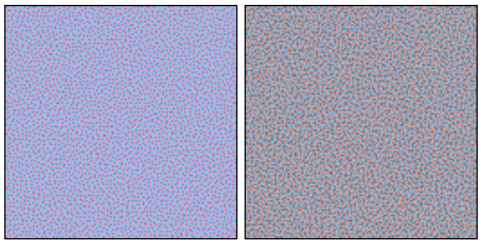

Fig. 13. Halftoned NPac created from simple (left) and more complex (right) NP stacking algorithm.

Working through the example, we start with: [C = 60%, M = 60%]. This exceeds 100%, by 20%, which is assigned to CM to give: [CM = 20%, C = 40%, M = 40%] (Fig. 13 (left)).

A more complex example (Fig. 13 (right)) starts with: [K = 30%, C = 50%, M = 50%, Y = 30%]. Here the sum begins at 160%. Starting with the last colorant, Y is 30% in excess of our maximum of 100%. So, 30% Y is joined with another NP, next in line M, giving: [K = 30%, MY = 30%, C = 50%, M = 20%]. The sum is reduced to 130%. Following the ordering, next, we take 20% M and join it with C, resulting in: [K = 30%, CM = 20%, MY = 30%, C = 30%] The sum drops to 110%, and we are left with an excess of 10% C. This 10% C is joined with K (the only remaining single ink NP), giving the final NPac: [CK = 10%, K = 20%, CM = 20%, MY = 30%, C = 20%].

This algorithm optimizes for coverage and minimum-ink NPs. Other optimization goals may be used to drive the decisions that create the NP probabilities.

III. PARAWACS Properties & Evaluation

Here we look at the property of plane dependence in PARAWACS, which refers to the halftone patterns of multiple colorants being such that the placement of one colorant informs the placement of another so that the combined spatial structure of the halftones of multiple colorants is independent of what colorants constitute it. If there is no plane dependence, the placement of one colorants is unaware of the placement of others and interference may result. Instead, if there is full plane dependence, then the combined halftone patterns of multiple colorants are like the pattern that would be obtained if only a single colorant were used.

To test the plane dependence of PARAWACS, a simple, two-ink, C and M system will be used here with the aim of getting 20% C and 20% M coverage and comparing it to a 40% C-only coverage. Two patches will be halftoned in each case: one, where each pixel contains the following NPac: [Blank = 60%, C = 40%] and the other where the NPac at each pixel is [Blank = 60%, C = 20%, M = 20%].

Using PARAWACS, the choice of NP at a pixel is driven by a threshold matrix, whose value for that pixel is compared to the cumulative area coverage vector at that same pixel. I.e., here we take the above NPacs and re-express them cumulatively as [W = 60%, C = 100%] and [W = 60%, C = 80%, M = 100%]. In other words, we look at how much area is covered by all NPs up to and including a given NP (which implies a fixed ordering). Then, if the current cumulative value is greater than or equal to the threshold value, the corresponding NP is chosen.

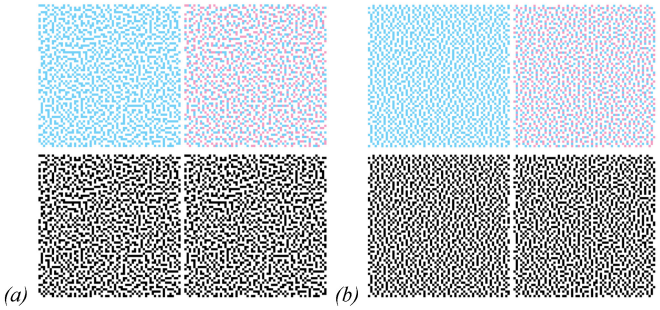

Fig. 14. 40% C versus 20% C plus 20% M (a) PARAWACS versus (b) error-diffused halftones.

Fig. 14a shows first the C-only result, then the C and M result, followed by a 64x64 detail and its B&W version of the two patterns. From here it can be seen that the two are identical (which is confirmed by subtracting the two images).

The reason for this identity between the W patterns in the C versus C and M cases is inherent in PARAWACS where a cumulative NP probability can be thought of as intersecting a threshold matrix and the cumulative probability values serving as break-points. Given a threshold matrix, it is the cumulative value (e.g. 40%), regardless of its underlying granularity (how many NPs add up to 40%), that solely determines the resulting halftone pattern. In the above case of [W = 60%, C = 100%] and [W = 60%, C = 80%, M = 100%] the cumulative value is the same at 60% for blank media, hence all values from the threshold matrix that go up until and including 60% - 153 in an 8bit range - will be left blank since they correspond to the blank substrate NP. The only difference is that in the [W = 60%, C = 100%] case, all values from [154 to 255] will contain the NP of one drop of C, while in the case of [W = 60%, C = 80%, M = 100%] values from [154 to 204] will be C while [205 to 255] will be M. However, in this second case too, non-blank NPs are placed at all values between [154 and 255] as before, thus resulting in the same pattern. So, all NPacs that have X% of blank substrate specified (assuming order is maintained and w is always the first NP) will have the same patterns with respect to blank. More generally still, any NPacs that have X% of NP Y will have the same patterns with respect to NP Y (assuming NP Y is either first or last in order).

For comparison, Fig. 14b shows the same NPacs error-diffused. As can be seen, the combined C and M pattern no longer matches the C-only pattern, but (and this is a consequence of HANS) there are no CM overprints as only C and M NPs were specified by the separation. The reason for there not being a strict match is twofold: first, there is an element of randomness in the error diffusion shown above and second, even when that randomness is turned off (and a more unpleasant pattern results) there is no match, since the error for the non-blank part of the pattern is in one case split between two other NPs (C and M) while in the other case it belongs all to one NP (C) and therefore has different 'granularity.' E.g., let's take the case of [C = 20%, M = 20%, W = 60%]. Here the first NP is blank and we propagate an error of [20 20 −40]. For the pixel to the right, its NPac will be the scaled

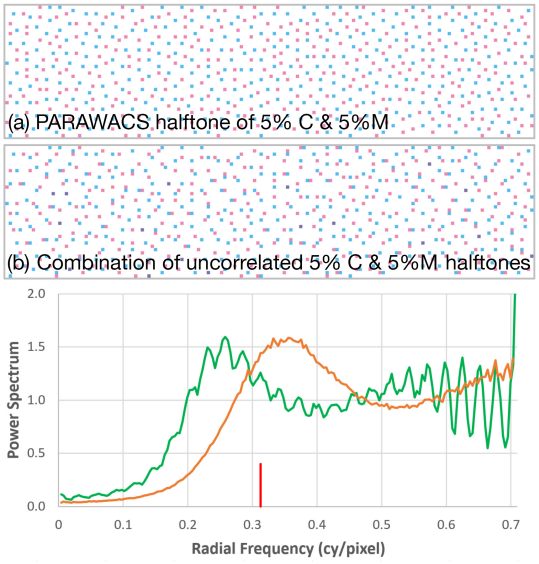

Fig. 15. Comparison of radially average power spectra of two halftones. (a) shown in orange, and (b) shown in green.

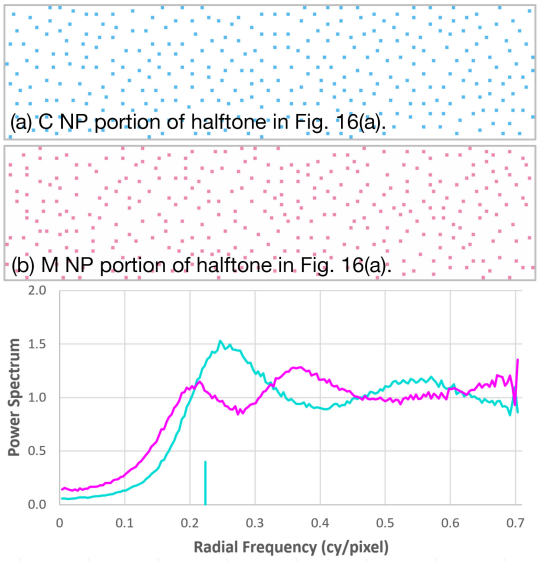

Fig. 16. Radially averaged power spectra of the above patterns. (a) cyan, and (b) magenta.

error (7/16)*[20 20 −40] + [20 20 60] which is [28.75 28.75 42.5] and we place w again. Now take the other NPac: [C = 40%, W = 60%]: the first NP is w and we propagate an error of [40 −40]. The pixel to the right's NPac will be the scaled error (7/16)∗[40 −40] + [40 60] which is [57.5 42.5] and we place C. The remainder after blank grows more quickly when applied to a single NP than if it is split between two.

IV. Frequency Domain Analysis

The complex nature of dispersed-dot halftones patterns can be simplified in the frequency domain. The Radially Averaged Power Spectrum [13] is a convenient one-dimensional signature of such patterns. Consider the case of generating of a fixed NPac of 5% C and 5% M by two different methods. Figure 15(a) shows the result of using PARAWACS and a blue noise matrix. By way of comparison Figure 15(b) shows an existing widely used method of halftoning that simply uses a threshold matrix to process each ink independently. In this case a blue noise threshold matrix and a circularly shifted copy of that matrix were used to binarize a 5% C image and a 5% M image, then combined to form the resulting uncorrelated halftone. The pattern is not nearly as homogeneous as that generated by PARAWACS in (a).

The radially average power spectrum of each plot is also shown in Figure 15 of the White NP of those patterns. This is equivalent to the aggregate pattern of all non-white pixels in the pattern. To arrive at these plots we first estimated the power spectrum by averaging 16 different samples of the pattern, each of size 256×256 pixels. We then segmented the spectrum into concentric annuli around the DC term and averaged the values within each annulus. These annuli extend to the corners of the baseband, which are at $\sqrt{2}$ cycles/pixel, so the plot covers the full range of frequency from 0 to .707 cy/pixel.

The PARAWACS (orange) curve in Fig. 15 exhibits a blue noise shape with energy concentrated at the principal frequency, white noise above the principal frequency and very little energy below the principal frequency. The red vertical line on the plot marks the principal frequency for the pattern shown. The nature of the blue noise principal frequency was first derived in [13] and revised in [14].

Starting with the same sample used in Fig. 15(a), we isolated the sub-images consisting of the single C-only and M-only NPs, and looked at their radially averaged power spectra. A small sample of the M-only and C-only images are shown in Fig. 16, along with its power spectra. The homogeneous arrangement of (b) the cyan pattern is reflected in the well-formed blue-noise spectrum. The magenta pattern (b), while well formed, is not quite as homogenous as the cyan-only image and this is manifested in the radially averaged power spectrum as a second peak above the principal frequency. It is important to point out that this pattern was not generated as a stand-alone M-only NPac halftone, but as part of the pattern from Fig. 15(a). Here too NP-order plays an important role. Exchanging the order of the C and M NPs would reverse the analysis. The reason for differences in spectrum is due to different cuts of the matrix having been used for the respective NPs.

The uncorrelated halftone in Fig 15(b) has considerably more low frequency energy below the principal frequency than in the PARAWACS spectrum. Low frequency energy corresponds to noisier visual appearance. Its spectrum is also not as smooth. However, the power spectra of the M-only and C-only component shown in Fig. 17 is homogeneous with a well-formed blue noise power spectrum. Because these two patters are generated with a circularly shifted version of the same threshold matrix, their power spectra are identical. Because the C and M patterns are uncorrelated, their sum results in somewhat random overlaps of C and M pixels. Those CM pixels in Fig. 15(b) are isolated and shown in Fig. 18. Interestingly, the power spectrum for this pattern is flat, corresponding to a white noise nature of this CM pixel pattern.

The shape of the spectrum shown as the green plot in Fig 16 can be explained from the duality of the spatial and

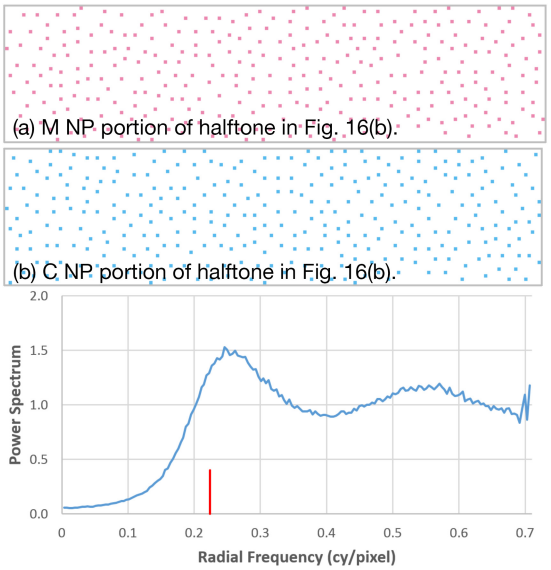

Fig. 17. Radially averaged power spectra of the above patterns. Both the magenta and cyan spectra are identical. (a) The Magenta NP portion of the halftone in Fig 16(b). (b) The Cyan NP portion of the halftone in Fig 16(b).

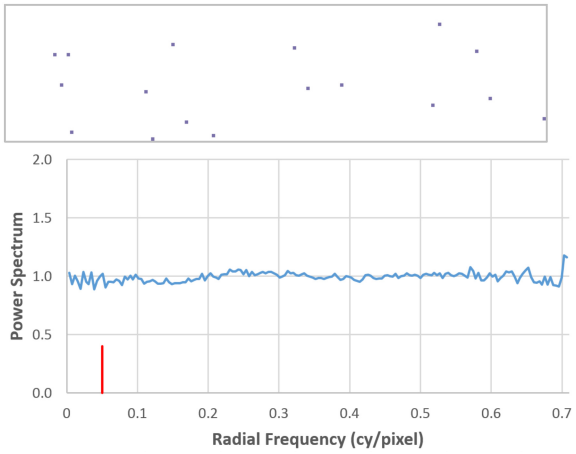

Fig. 18. The CM NP portion of the halftone in Figure 16(b), and its radially averaged power spectrum.

frequency domains. Since the C and M patterns are shifted versions of shifted versions of each other, in the spatial domain it can be described as a single pattern convolved by pair of delta functions, one for the original and one for the shifted version. In the frequency domain this corresponds to the spectrum of one pattern multiplied by a 2D sinusoid.

One measure of the radial symmetry in halftone spectra is anisotropy, which can be defined as the sample variance within the concentric annuli as a function of frequency, summarized as a one-dimensional plot. All of the halftone patterns presented in this section demonstrate good radial symmetry except for the uncorrelated pattern in Fig 15(b). The difference in radial symmetry of the two halftones in Figure 16 is best illustrated in a full 2D plots as shown in Fig 19. The PARAWACS plot in Fig 20(a) is smooth and symmetric about the DC origin. However, the uncorrelated plot in (b) reveals the sinusoidal interference. Because we used a circular shift of [25, 25] of our 1024×1024 blue noise threshold matrix in that case,

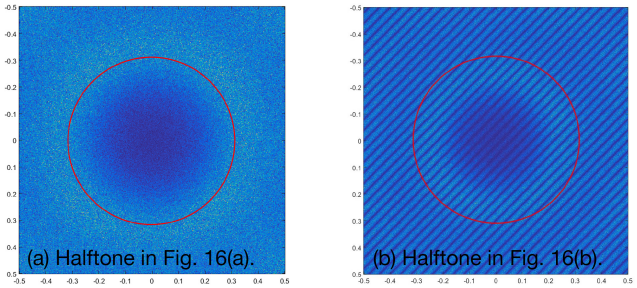

Fig. 19. Comparison of the full 2D Fourier transform magnitudes of the halftones from Figure 16. The principal frequency is shown as a red circle. (a) The halftone in Fig 16(a). (b) The halftone in Fig 16(b).

the 2D sinusoid is oriented along a 45-degree angle as shown. This directional sinusoid imparts anisotropy in the spatial pattern and describes the ripples in the radial spectrum shown as the blue plot in Fig. 15. While thresholding individual inks with a blue noise matrix produces well-formed homogeneous halftones for each ink, the homogeneity is not preserved in the sum of such patterns. This is unlike the case of PARAWACS where a single matrix determines the pattern of the entire ink-stack (including its overprints determined a-priori).

V. Conclusions & Future Work

Color halftoning impacts many aspects of a print, such as grain, smoothness and color itself. A novel, predictable, deterministic algorithm was described. It was shown to give a great degree of control over final output patterns and is well behaved in that for different content that shares the overall coverage, the halftone pattern is provably constant.

An important consequence of PARAWACS is that there are more critical choices to be made when designing halftone matrices, since their result directly impacts the final print, unlike in other matrix based halftoning approaches where the impact is less immediate and matrices have to work jointly towards a final halftone result. Any research into patterns that are visually pleasing or exhibit particular behavior can directly be applied, as can methods for their evaluation (e.g., [20]), making PARAWACS a uniquely flexible approach.

We presented the basic principles of PARAWACS and its application to general purpose halftoning. Future work will consider applying additional metrics to halftones generated using PARAWACS, including an exploration of its impact on moiré in printed halftones and analysing the variance of Neugebauer equations as a function of window size. The level of control PARAWACS provides will also be explored in security printing, where embedding particular patterns in a single matrix directly translates to a change in the overall halftone pattern [15]. A natural extension to 3D printing will also be followed up, where instead of a 2D matrix of particular properties a 3D native matrix can be designed, with control over 3D patterns in general and layer-to-layer interactions, connectedness, porosity, micro and macro geometry in particular.

Acknowledgements

The authors would like to thank Tsuyoshi Yamashita for providing the halftone matrices used throughout this paper and

Utpal Sarkar for the matrix used in Fig. 9. They would also like to thank the reviewers for their helpful comments and the following colleagues at HP Inc. for their support: Annarosa Multari, Africa Real, Rafa Gimenez, Albert Serra and Ramon Pastor.

REFERENCES

[1] I. Amidor, R. D. Hersch, and V. Ostromukhov, "Spectral analysis and minimization of Moiré patterns in colour separation," *J. Electron. Imag.*, vol. 3, no. 3, pp. 295–317, 1994.
[2] J. S. Gondek, "Correlating cyan and magenta planes for error diffusion halftoning," U.S. Patent 5 949 965, Sep. 9, 1999.
[3] J.-H. Lee, M. Schramm, and J. Quintana, "Sequential color error diffusion with forward and backward exchange of information between color planes," U.S. Patent 7 440 140, Oct. 21, 2008.
[4] Y. Zhang, J. L. Recker, R. Ulichney, and J. Owens, "Plane-dependent error diffusion on a GPU," *Proc. SPIE Image Process., Algorithms Syst. Parallel Process. Imag. Appl.*, vol. 8295, p. 829515, Feb. 2012.
[5] J. Morović, P. Morović, and J. Arnabat, "HANS: Controlling ink-jet print attributes via neugebauer primary area coverages," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 688–696, Aug. 2011.
[6] R. Ulichney, *Digital Halftoning*. Cambridge, MA, USA: MIT Press, 1987.
[7] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green-noise digital halftoning," *Proc. IEEE*, vol. 86, no. 12, pp. 2424–2444, Dec. 1998.
[8] R. A. Ulichney, "Void-and-cluster method for dither array generation," *Proc. SPIE Human Vis., Vis. Process. Digit. Display*, vol. 1913, p. 332, Sep. 1993.
[9] J. P. Allebach and Q. Lin, "FM screen design using DBS algorithm," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1. Lausanne, Switzerland, Sep. 1996, pp. 549–552.
[10] H. Bohr, "Zur theorie der fast periodischen funktionen: I. Eine verallgemeinerung der theorie der fourierreihen," *Acta Math.*, vol. 45, pp. 29–127, 1925.
[11] M. E. Demichel, "Le procédé," vol. 26, pp. 17–21, 1924.
[12] M. T. Schramm and J. S. Gondek, "Plane dependent matrix based halftoning," U.S. Patent 6 870 645, Mar. 22, 2005.
[13] R. A. Ulichney, "Dithering with blue noise," *Proc. IEEE*, vol. 76, no. 1, pp. 56–79, Jan. 1988.
[14] D. L. Lau and R. Ulichney, "Blue-noise halftoning for hexagonal grids," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1270–1284, May 2006.
[15] P. Morovic, J. Morovic, J. Gondek, M. Gaubatz, and R. Ulichney, "PAWARACS: Color halftoning with a single selector matrix," in *Proc. 24th Color Imag. Conf.*, Nov. 2016, pp. 7–11.
[16] J. Lee and J. P. Allebach, "Colorant-based direct binary search halftoning," *J. Electron. Imag.*, vol. 11, pp. 517–527, Oct. 2002.
[17] Z. He, "Hierarchical error diffusion," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1524–1535, Jul. 2009.
[18] Z. He, "Hierarchical colorant-based direct binary search halftoning," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1824–1836, Jul. 2010.
[19] Y. H. Fung and Y. H. Chan, "Blue noise digital color halftoning with multiscale error diffusion," *J. Electron. Imag.*, vol. 23, no. 6, p. 063013, 2014, doi: 10.1117/1.JEI.23.6.063013.
[20] H.-W. Chang, H. Yang, Y. Gan, and M.-H. Wang, "Sparse feature fidelity for perceptual image quality assessment," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 4007–4018, Oct. 2013.
[21] J. B. Rodriguez, G. R. Arce, and D. L. Lau, "Blue-noise multitone dithering," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1368–1382, Aug. 2008.
[22] G. Sarailidis and I. Katsavounidis, "A multiscale error diffusion technique for digital multitoning," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2693–2705, May 2012.

Peter Morovič received the B.Sc. degree in theoretical computer science from Comenius University, Slovakia, and the Ph.D. degree in computing sciences from the University of East Anglia in 2002. He was a Senior Research Fellow with the University of Chiba and a Senior Research Associate with the University of East Anglia. Since 2007, he has been a Senior Color and Imaging Scientist with HP Inc. He has authored over 50 scientific articles and his research interests include color vision, color reproduction, 2D/3D image processing, and computational geometry. He is a member of the CIE Technical Committee 8-07 on multispectral imaging. He received the IS&T's Raymond Davis Scholarship in 2001 and a JSPS Post-Doctoral Fellowship in 2005.

Ján Morovič received the B.A. degree (Hons.) in print management from the London College of Printing in 1995 and the Ph.D. degree in color science from the University of Derby in 1998. Between 1998 and 2003, he was a Lecturer with the University of Derby. Since 2003, he has been with HP Inc. He was the past Director of the CIE Division 8 on Image Technology from 2011 to 2014. He has authored over 100 scientific papers and the book *Color Gamut Mapping* (John Wiley and Sons, 2008). His research interests include color science and color reproduction. He was a recipient of the Royal Photographic Society of the Great Britain's 2003 Selwyn Award.

Jay Gondek received the B.S. and M.S. degrees in computer science from the University of Oregon in 1994 and 1992, respectively. He joined HP Inc. in 1994 and has contributed to a wide gamut of imaging products ranging from cameras and displays to consumer, office, retail, and professional color printers. He is currently an Distinguished Technologist for color and imaging systems with HP Inc. He holds over 35 patents in the areas of halftoning, color reproduction, image enhancement, and printing. His most recent released projects include PageWide inkjet printers and the Sprocket consumer photo printer.

Robert Ulichney (M'80–SM'88) received the B.S. degree in physics and computer science from the University of Dayton, OH, in 1976, and the M.S., E.E., and Ph.D. degrees and the M.A. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1979, 1984, and 1986, respectively. He was with Digital Equipment Corporation and then with the Compaq's Cambridge Research Lab. He is currently a Distinguished Technologist with HP Labs, USA, involving in systems for high-capacity data embedding, and structures for variable density 3D printing. He has over 50 issued patents and 70 technical publications in a range of imaging areas. He is a fellow with IS&T.