

# PARAWACS: color halftoning with a single selector matrix

Peter Morovič, Ján Morovič, Jay Gondek, Matthew Gaubatz, Robert Ulichney, HP Inc., Spain & USA

## Abstract

Halftoning is one of the key stages 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 the individual colorants, which becomes increasingly complex and over-constrained as the number of the colorants increases. Both choices of screen angles (in clustered-dot cases) or structures and control over how the individual matrices relate to each other and result in over- versus side-by-side printing of the colorants impose restrictions that are challenging to reconcile. The solution presented in this paper relies on the benefits of a HANS 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 used. The provably complete plane-dependence of the resulting halftones and an application to security printing will be presented among the solution's benefits.

## 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 the 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 then responsible for making spatial choices of where to apply each colorant in turn, given the choices of colorant amounts made during halftoning. Since halftoning colorant amounts one by one can lead to unwanted interactions between their halftone patterns, previous work has looked at making choices of individual halftoning matrices that minimize moiré effects (e.g., Amidor *et al.*, 1994), or that provide plane-dependency between specific pairs of colorants, allowing for their placement to be kept apart for as long as possible (e.g., Zhang *et al.*, 2012).

Since such color halftoning challenges follow from the difficulties 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 presented to halftoning. This is precisely what the introduction of HANS (Morovič *et al.*, 2011) 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, CMY, where color separation previously specified continuous amounts of CMY, a HANS pipeline specifies relative area coverages for the system's 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).

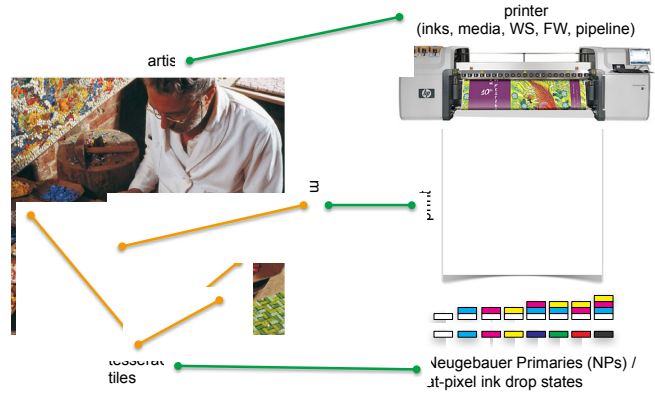


Fig. 1: HANS print control as mosaic building.

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 the cyan colorant and 10% contain the combination of the magenta and yellow colorants.

In previous work, the halftoning applied to such NPacs has been error-diffusion based (Morovič *et al.*, 2011), 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 cyan and magenta plus yellow being short by 30% and 10% respectively (Fig. 2). This error would then be diffused to neighboring pixel's NPacs using existing error diffusion techniques (Ulichney, 1987).

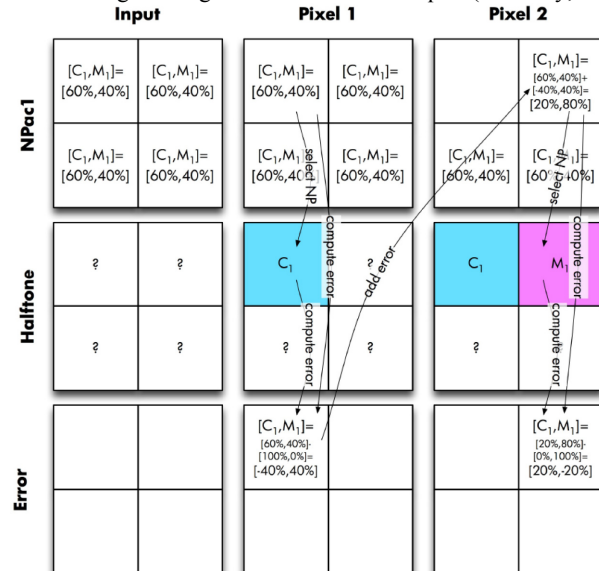


Fig. 2: HANS error diffusion example.

While error diffusion is an excellent way for taking NPacs determined by a color separation and constructing colorant patterns from them, it does have several limitations. The first of these is throughput, since a sequential traversal of halftone pixels is required and even though there have been some ingenious attempts at parallelizing error diffusions (Zhang *et al.*, 2012), even their speeds are nothing like 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 error diffusion is that it involves 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 requires deterministic patterns (e.g., the stitching together of halftones for separate parts of a printing system, or the application of data embedding).

Both of these challenges point to the use of matrix-based halftoning for HANS. However, existing solutions, where there is potentially a different halftoning matrix for each colorant and where halftoning is done mostly independently for separate colorant channels, are not directly applicable to HANS. The key reason for this is the difference in the two domains – the colorant one and the NPac 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 combination statistics. As a result, the separate use of halftoning matrices is not an option, since their application to an NP-by-NP basis would introduce even greater constraints on their generation that colorant channels do and since the number of NPs can exceed several hundred or even thousand, this may not even 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 an illustration of some of its benefits by applying it to security printing.

## 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 of the unit area that need to be occupied by each of the NPs present in the NPac, at the proportions given by the area coverages. Hence, one way to think about the role of halftoning in this context 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 NP area coverage is obtained. Note that 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 x 128 pixels that has a constant NPac of 80% blank substrate, 10% one drop of Magenta ink and 10% of Cyan ink, then if halftoning just sequentially placed them over the specified area, the patch would look like that in Fig. 3.

Counting the number of pixels in the above halftone will yield 1638 pixels (10%) of each of Magenta and Cyan and the remaining

pixels 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.



Fig. 3: A halftoned patch of a constant NPac of 80% blank media, 10% of one drop of magenta and 10% of one drop of cyan over an area of 128 x 128 pixels.

Another way to think about the example above is the following: given a (sufficiently large) area of a constant NPac such as that shown in Fig. 3, its halftone should have the following property: *The likelihood of picking one of the NPs from the NPac 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 Figure 1 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 does not have another important attribute of halftoning, which is a uniform spatial distribution of the NPs, instead, it simply clustered and ordered all states sequentially.

Since this is a probability distribution, a simple way to have a more uniform distribution is to simply generate uniformly distributed random numbers, using a standard random number generator and scaling them to a range of 0 to 100. 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 Magenta 10%, 1 drop Cyan 10%]  
becomes

[Blank 80%, 1 drop Magenta 90%, 1 drop Cyan 100%]

which in turn defines intervals for each of the states such that [0 to 80] corresponds to the Blank state, [80 to 90] to one drop of Magenta and [90 to 100] to one drop of Cyan. Given this representation, the random numbers generated then simply need to be categorized according to the intervals. If a random number at [x, y] is in the range [0 to 80] it is left blank, if it is in the range [80 to 90] a one drop of Magenta NP is placed and if the random number is in the interval [90 to 100] a Cyan drop NP is placed. The diagram below shows this process for four random values (corresponding to four [x, y] locations):

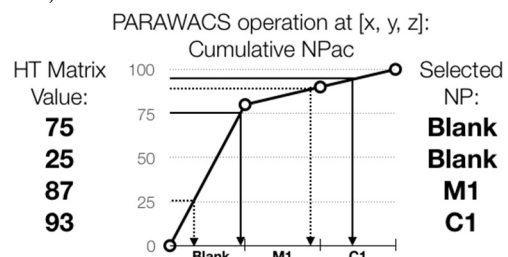


Fig. 4: Uniform random numbers used to halftone the same NPac as shown in Fig. 1.

And Fig. 5 below shows the result of applying this process based on a uniform random number generator:

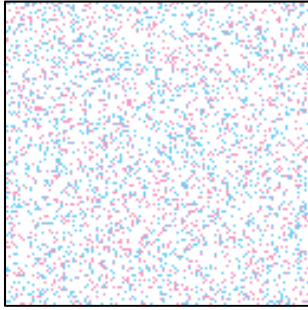


Fig. 5: Uniform random numbers used to halftone the same NPac as shown in Fig. 3.

The halftone shown in Fig. 5 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 (Lau *et al.*, 1998), 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 over the overall resulting halftone pattern. And while Fig. 5 is significantly better than Fig. 3 in terms of how evenly the NPs are placed, 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 x 128 pixel area.

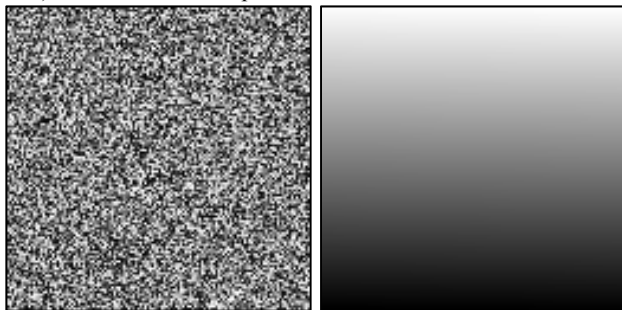


Fig. 6: A grayscale visualization of the uniform random numbers used to produce the halftone in Fig. 5 (left) and of an ordered, sequential choice of values used for the halftone in Fig. 3 (right).

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 terms as well.

The values in Fig. 6 are exactly the same 128x128 grayscale values, except they are placed differently – in one case in a uniform random manner and in the other case simply sorted by grayscale value and placed sequentially.

An important property of PARAWACS halftoning can be seen here already: the spatial nature of the values used in the selection of NPs from the NPac 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 (Ulichney, 1993). An example of clustered-dot halftoning where clusters are built on

blue noise centers is referred to green noise (Lau *et al.*, 1998). Figs. 7 and 8 show an example of the blue and green noise arrangements – both the halftone matrices as well as the result when applied to the same constant NPac patch as shown in Figs. 3 and 5.

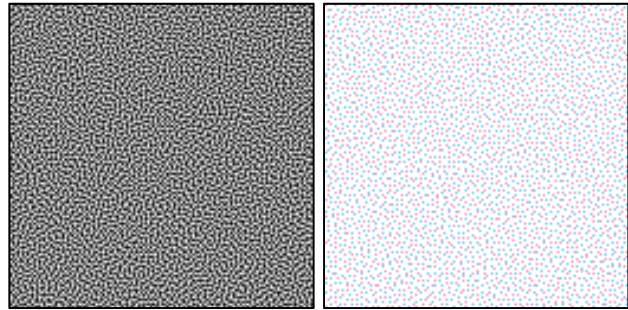


Fig. 7: A blue noise halftone matrix (left) and the resulting halftone (right).

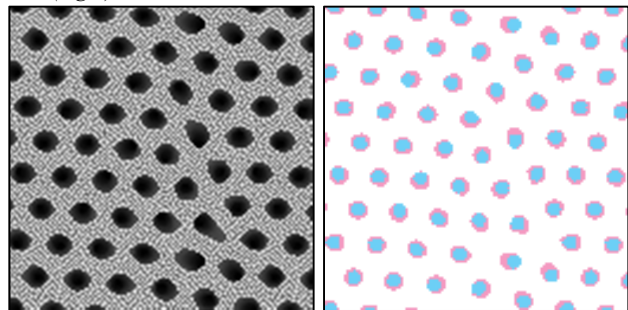


Fig. 8: A green noise halftone matrix (left) and the resulting halftone (right).

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 Magenta (10%) and the same number of pixels occupied by one drop of Cyan (10%), but they differ in the spatial arrangement of the halftone values (the halftone matrix). Hence the above halftone matrices will have the property of having a uniform probability of each of the gray-scale values: e.g. using an encoding of [0 to 255] for the grayscale values there will be  $(128 \cdot 128) / 255$  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 (or random numbers for the white-noise case) are generated as well as the patch size over which an NPac is constant determine the accuracy with which an input NPac can be reproduced.

Another parameter that influences the final halftone pattern is that of the order of NPs in an NPac, based on which the cumulative distribution is built. Whether blank substrate is first or last determines which 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 below where the same green-noise matrix is used as in Fig. 8, except reordering the NPs in reverse order (one drop of cyan first, then one drop of magenta 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 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 traditional ink-channel based halftoning where the ink-recipe only determines ink 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 Cyan and one drop of Magenta placed at the same  $[x, y]$  location is an indirect consequence of halftoning. This is both true for traditional matrix-based halftoning and error-diffusion since both act in ink-channels. While it is possible to correlate (or de-correlate) ink 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 possible concurrently in a single pipeline and is guaranteed to be maintained in the halftone.

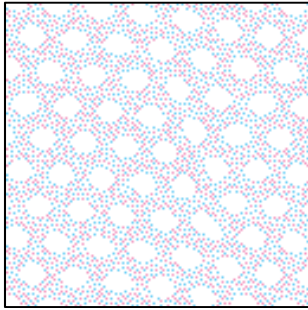


Fig. 9: A green noise halftone using the same halftone matrix from Fig. 8 (right), changing the order of NPs in the NPac.

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).

While in the examples used to illustrate the halftoning mechanism a square patch has been used, this probabilistic understanding of NPacs and their halftoning also applies at a single pixel. As is the case in general image content, every pixel of an input image can vary in terms of its RGBs and therefore once a HANS color separation is applied, can have a different NPac. Hence NPacs are better thought of as a probabilities rather than a recipes since at any one pixel a single NP needs to be selected from among the constituent 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 therefore 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 ink-channel recipes at each pixel that a color separation determines are, in effect, 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 that location.

Finally, to illustrate the halftoning algorithm on more complex content a 1024x1024 pixel 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 been found to be most pleasing for image content.



Fig. 10: An 1024x1024 image halftoned using three different halftone matrices: white noise (top), blue noise (middle), green noise (bottom).

## Plane dependence

Plane dependence 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 what colorants constitute it. If there is no plane dependence, the placement of one colorant 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 is like the pattern that would be obtained if only a single colorant were used and that would therefore have the desired spatial properties.

To test the plane dependence of PARAWACS, a simple, two-ink, cyan and magenta system will be used and the aim will be to get 20% C and 20% M coverage and compare it to a 40% C-only coverage. Therefore, two patches will be halftoned in each case: one, where each pixel contains the following NPac:  $[w=60\%, c=40\%]$  and the other where the NPac at each pixel is  $[w=60\%, C=20\%, M=20\%]$ .

Using PARAWACS, the choice of NP at a pixel (from among the non-zero NPs specified for it in the input) 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 of NPs). Then, if the current cumulative value is greater than or equal the threshold value, the corresponding NP is chosen.

Fig. 11 then shows first the C-only result and 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 from subtracting the two images from each other, which yields a zero image).

The reason for this identity between the W patterns in the C versus C and M cases is inherent in the nature of 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% - a value of 153 on 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 Cyan, while in the case of [w=60%, C=80%, M=100%] values from [154 to 204] will be Cyan while [205 to 255] will be Magenta. 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).

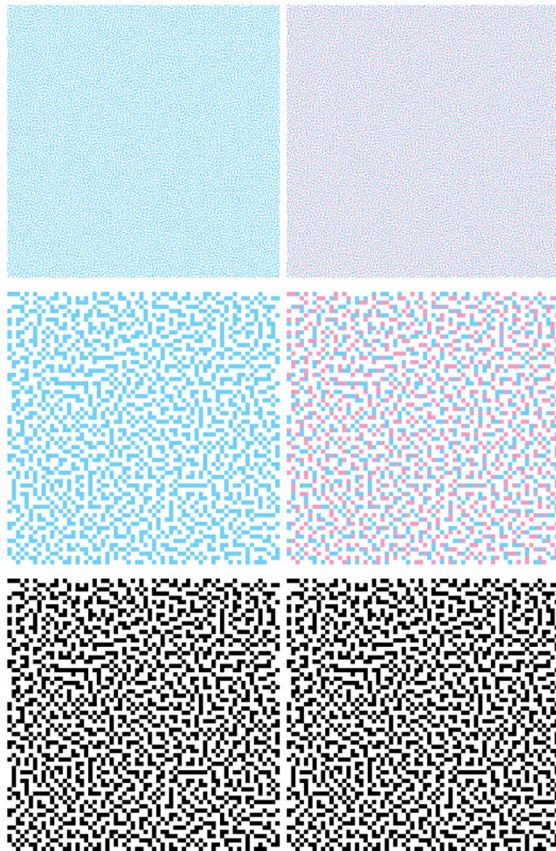


Fig. 11: 40% cyan versus 20% cyan plus 20% magenta PARAWACS halftones.

For comparison, Fig. 12 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 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 than if it is split between two.

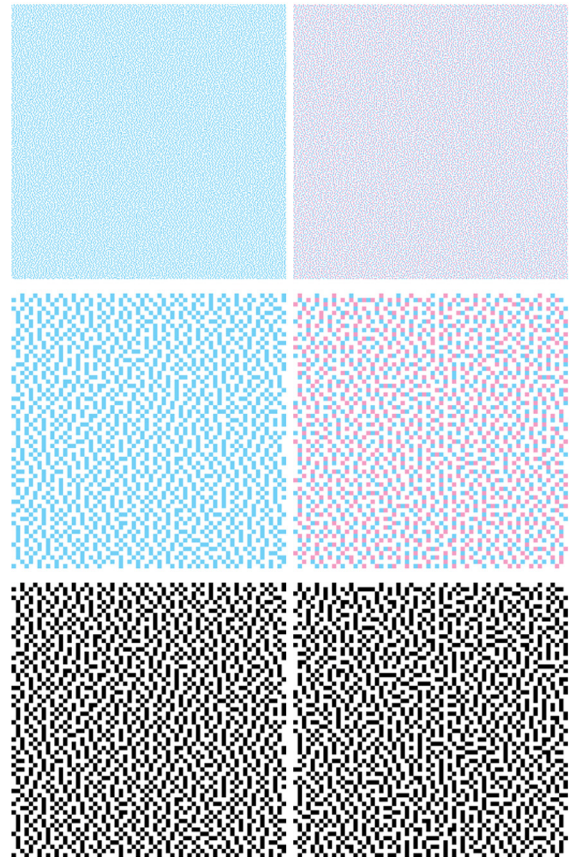


Fig. 12: 40% cyan versus 20% cyan plus 20% magenta error-diffused halftones.

## Security Printing

As has been shown, PARAWACS preserves patterns characteristic of the halftone matrix and does so independently of the NPacs, their composition (which NPs are used at what coverage) or complexity (how many different NPs are used), being halftoned. This follows directly from the halftoning algorithm described earlier, which uses the halftone matrix as a point-wise selector of device states or NPs. These properties lend themselves naturally to security printing in the form of inducing intentional structure into the content. The goal of doing so may include an implementation of covert or overt watermarking, data embedding, authentication, tracking, anti-counterfeiting, etc.. As security features, modifications of halftone structure offer a convenient blend of fragility to counterfeit production processes and reliability of detection.

The PARAWACS approach offers some advantages over other security printing techniques. While watermarking for print is more commonly performed on continuous tone data (Kacker *et al.*, 2003), the idea to manipulate halftones directly in order to embed unique patterns has been explored before (see Chiang *et al.* 2009 for a survey of such methods). However, these methods typically operate on binary (black and white) halftones and are not suitable to extend to full color content directly. They also often involve content-dependent processing or post-processing of the conventionally halftoned content. Instead, what is novel to the PARAWACS approach is that no image-dependent manipulation or post-processing occurs; instead, the halftone matrix itself is modified with a particular spatial pattern and, once modified, it is used in the same way as shown earlier (with white, green, blue noise or any other halftone matrix). Thus the approach extends naturally to full color content. It can also be applied to *any* image content, unlike binary or post-processing techniques which typically only apply to dedicated regions of an image (and therefore indirectly indicate the content that is intended to be a covert pattern).

An illustrative example is formed motivated by methods in the literature that operate via modifying halftone patterns. Stegatones, for instance, replace windows of a halftone with carefully chosen, rearranged versions of the contents of the same window (Ulichney *et al.* 2010). For clarity, a simpler modification is used here, applied directly to the halftone matrix: values within  $N \times M$  windows of the halftone matrix values are rearranged into ascending order. This method clearly modifies the spatial characteristics of the halftone matrix, without affecting the frequency of each halftone value, thus maintaining the basic constraint of a halftone matrix and only altering its local spatial arrangement.

For a given blue-noise matrix, and modification window sizes of  $2 \times 2$  and  $8 \times 1$ , the locally-sorted halftone matrices are shown in Fig. 13. Applications of these matrices to a test photograph in Fig. 10 are in Fig. 14. Note that the more pronounced the patterning, from the  $8 \times 1$  windowing with the simplistic sorting operation, results in a more visible impact on the image quality while the  $2 \times 2$  windowed version differs less visually from the default blue noise version in Fig. 10 while still having a significantly different microstructure. Clearly, it is possible to induce different spatial behaviors by shifting halftone values around, which in turn results in detectable and even visible alterations of an image.

The examples illustrate that a spatial modification of an existing halftone matrix, or a custom design of one which still satisfy the constraints of a valid halftone matrix (such as having uniform frequency of each of the halftone values), is directly applicable for halftone-independent, full color content. Much of the work done in literature on pattern embedding or watermarking at halftone level can be leveraged in this context, by building halftone matrices for PARAWACS that have properties like those of the final, binary halftones using traditional methods. Note the degree to which the embedded pattern will transmit to a halftone is variable, since no particular region of an input image needs to be dedicated as pattern-bearing region, although this, too, is possible with the proposed approach. Detection schemes may thus operate on larger image regions or involve continuous capture to sample appropriate content.

## Conclusions

Color halftoning impacts many aspects of a print, such as grain, smoothness and color itself. A novel, predictable, deterministic algorithm was described that has been shown to give a great degree of control over the final output patterns and is well behaved in that for

different content that shares the overall coverage, the halftone pattern is provably constant. As a result, it is also well suited for applications such as security printing for watermarking or data carrying.

An important consequence of PARAWACS is that there are more and more important 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. Any research into visually pleasing or particular behavior exhibiting patterns can directly be applied, making PARAWACS a uniquely flexible approach.

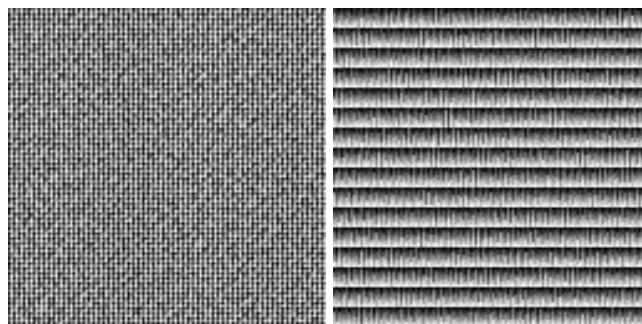


Fig. 13: Locally sorted halftone matrices based on the blue noise matrix in Fig. 7 (left) using a windowing operation of  $2 \times 2$  (left) and  $8 \times 1$  (right).



Fig. 14: Test image halftoned using the two modified halftone matrices from Fig. 13 with windowing operations of  $2 \times 2$  (left) and  $8 \times 1$  (right). A  $32 \times 32$  pixel corner is enlarged by  $10 \times$  to see the detail.

## References

1. Amidor, R. D. Hersch, V. Ostromukhov "Spectral Analysis and Minimization of Moiré Patterns in Colour Separation," *Journal of Electronic Imaging*, 3(3), 295-317, (1994)
2. R. Ulichney, "The Void-and-Cluster Method for Generating Dither Arrays", *Human Vision, Visual Processing, and Digital Display IV*, Proc. SPIE 1913, pp. 332-343, (1993)
3. D. L. Lau, G. R. Arce, N. C. Gallagher "Green-noise digital halftoning" *Proceedings of the IEEE*, 86(12):2424 - 2444 (1998)
4. J. Morovič, P. Morovič, J. Arnabat "HANS – Controlling Inkjet Print Attributes Via Neugebauer Primary Area Coverages," *IEEE Transactions on Image Processing*, 21, 2, (2011)
5. R. Ulichney "Digital Halftoning," MIT Press, Cambridge, MA, USA (1987).
6. Y. Zhang, J. L. Recker, R. Ulichney, J. Owens "Plane-dependent Error Diffusion on a GPU," *SPIE Proceedings*, 8295:8295b-59:1-10, (2012)
7. D. Kacker and J. P. Allebach, "Joint halftoning and watermarking," *IEEE Trans. Signal Processing*, vol. 51, no. 4, pp. 1054–1068, Apr. 2003
8. P. Chiang, N. Khanna, A. Mikkilineni, M. Segovia, S. Suh, J. Allebach, G. Chiu, and E. Delp, "Printer and Scanner Forensics", *IEEE Signal Processing Magazine*, March, (2009)
9. R. Ulichney, M. Gaubatz, and S. Simske, "Encoding Information in Clustered-Dot Halftones", *IS&T NIP26 (26th Int. Conf. on Digital Printing Technologies)*, Austin, TX, 602-605, Sep 2010.