https://doi.org/10.2352/ISSN.2169-2629.2018.26.410

Halftone structure optimization using convex programming

! "#"$%& ' $' ()*+%, - . %& ' $' ()*+%/)0#' $%1)"2 ' +%, 3 ()"$%& 3"4#$' +%5 "0#' $%6 ' 7 "8+%9 "$2)' %: #0; "<" ; "$"+%= 3 ()"$%>3$)? 3+%! "$"%6 34@3$). %
5 ! %A 0B%9 3 . #%C D23#%E "F%/ 3F"4+%9 @3). %

! " # $%& ' $(

In a color printing pipeline, the overall properties of a printed pattern depend on how the available inks are used to reproduce a given color. These choices are typically made in ink-space and therefore concern only how much of each ink to use, with halftoning then determining how those ink amounts interact with each other. For a HANS (Halftone Area Neugebauer Separation) pipeline, it is already the color separation – the recipes of how inks are used for a given color – that determines not only ink amounts but also how to combine them, while halftoning only provides their spatial distribution. Hence, with HANS, halftone properties are to a larger extent determined already by the color separation. This is an important change from traditional pipelines and requires different methods of control. This paper describes an approach to achieving new levels of image quality performance, without incurring complexity in the pipeline. It is based on the realization that certain strategies of halftone composition result in less grainy prints, e.g., when minimizing the amount of blank substrate, and more generally minimizing contrast among constituent, at-pixel drop-states (ink combinations). This is achieved through the mathematical technique of convex optimization, where such strategies can be formulated and efficiently computed. Results are shown for node-by-node LUT optimization; calibrating LUTs from ink-channel ratio data; transforming LUTs to take into account changes in drop-sequences, sets of admissible NPs or the number of possible NPs to use in an on-line pipeline and generating color samples that match in color but vary in grain. This approach is deeply embedded in the first HANS commercial product, the HP DesignJet Z6 and Z9$^+$ series, a portfolio of pro-photo/graphics printers.

) &' *+%, - . / ((

Printing systems involve a broad variety of elements, processes and operations that together result in the systems' overall performance, such as its color consistency, color uniformity, print image quality and color preference. To deliver high levels of performance, it is therefore necessary to derive resources for a system's variety of color components, such as color calibration, color separation and color management, so that these may make the system perform to its full potential. Here it is necessary to reconcile the differing requirements and capabilities of such components, e.g., firing frequency constraints of print heads, overall ink limits and per-ink cut-offs, constraints of the hardware running the data path or the capabilities of color sensors, etc. To balance such a complex system, there is a constant need for a broader and more versatile toolset for building and adjusting resources and it is in this context that this paper presents a new mechanism that bridges the conventional ink-space domain and the HANS [1] paradigm to deliver a highly versatile way for taking system constraints and requirements into account.

With a move to the HANS domain, where color resources are built using Neugebauer Primary (NP) area coverages (NPacs) – all possible combinations of ink-drop states – instead of ink channels, a host of new choices and control mechanisms become available. At the same time, the ink-channel domain – which is the currently used domain – has led to high quality output and valuable technologies. For example, building color resources in ink-space is a radically simpler problem than doing so in NPac space due to the many orders of magnitude of difference: a 4-ink system with 3 drop states has 81 NPs, resulting in a 4D ink-space vs a 81D NPac space. Another example is calibration, where changes in drop-size can reliably be estimated from measurements of ink-ramps and applied directly in the ink-channel domain. Finally, high-end halftoning approaches, typically based on error diffusion, such as Tone-Dependent Error Diffusion (TDFED) [2, 3], result both in high-quality per-ink patterns and a tendency of favoring overprint states. These technologies, however, are not directly applicable to HANS since its NPac domain is in a many-to-one relationship to ink-channels, which is also the reason for HANS having access to new patterns. The challenge then is to reconcile the benefits of the new domain, while being able to leverage ink-channel native technologies or their insights. Finally, there are new, HANS-specific conditions as well. The two halftoning families developed for HANS so far: Device State Error Diffusion (DSED) [1] and Parallel Random Area Weighted Coverage Selection (PARAWACS) [4] both only serve the purpose of spatial distribution of the device states dictated by the NPacs. Particularly in the case of PARAWACS, simpler NPacs using fewer NPs at larger area coverages result in better patterns as they slice the single halftone matrix fewer times and into larger, contiguous chunks. For DSED, which operates in NP rather than ink space, keeping the dimensionality low has throughput benefits.

In what follows, an optimization framework is presented that allows for transforming NPacs to new NPacs, taking their ink-vectors into account, with the ability to introduce NPac property metrics. This framework enables an efficient solution of problems such as: combining ink-channel and NPac technologies for resource generation; calibrating a LUT via ink-amount change measurements; optimizing IQ for an existing LUT by introducing new properties; changing drop-sequences or sets of NPs of a LUT. In order to describe these applications, first the convex programing framework will be introduced, followed by a detailed description of each application as well as its results.

0 , . 123 4 %, + %&5 5 6. + 7%&5 28 , %* (

Let us first define the relationship between ink-vectors and NPac-vectors. An ink vector $\boldsymbol{o}_i$ for an i-th color in a set of colors, represents the amount of each of the available inks of a system, such that for CMY inks:

! " # $%& (%& (%&)

would correspond to 0.7 drops of Cyan ink, 0.7 drops of Magenta and no drops of Yellow on average per pixel over some area. An NPac vector instead is the area coverage or probability of each of the possible Neugebauer Primaries of the system. For a CMY ink-set and a binary deposition mechanism there are $M = 8$ NPs which can be ordered, for example, as follows: [w, C, M, Y, CM, CY, MY, CMY], with w representing blank pixels. Assuming this fixed order of NPs, for the same color i, an NPac vector $\boldsymbol{n}_i$ can then be written as[1]:

$$\boldsymbol{n}_i = [20, 10, 10, 0, 60, 0, 0, 0].$$

Meaning 20% of blank pixels, 10% of Cyan pixels, 10% of Magenta pixels and 60% of Cyan and Magenta pixels overprinting. The relation between ink-vectors and NPac vectors is a one-to-many relation and is determined by means of an NP matrix, $\boldsymbol{N}$, which in this example case is 8x3 (M = 8 NPs, 3 inks) and is referred to as the drop-state matrix where each NP is represented in a row. For example, one drop of Cyan combined with a drop of Magenta would be [1 1 0], with the full matrix, in the same order as before (starting with "w" and ending with "CMY"):

$$N^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

To compute the ink-vector from an NPac-vector is then simply a multiplication:

$$\boldsymbol{n}_i \cdot N = \boldsymbol{v}_i$$

The above relationship is well defined in one direction, however, to invert it is ill-posed: there are many possible NPacs for a given ink-vector. Naïve mathematical tools like pseudo-inverses or regression would result in a solution but this can have two problems: 1. the solution may not satisfy the definition of an NPac and 2. the properties of the NPac are arbitrary (e.g., in terms of grain). Hence, the next step is the introduction of a computational framework for solving the inverse problem such that the solutions (since there are many possible ones by virtue of the under-determined nature of the problem) satisfy the definition of NPacs and are therefore valid, and that their properties can be controlled and optimized.

First, since NPacs represent area coverage proportions or probabilities, they have to be convex, such that:

$$\forall \boldsymbol{n}_i[j]: 0 \le \boldsymbol{n}_i[j] \le 1$$

$$\sum_{j=1}^{M} \boldsymbol{n}_i[j] = 1$$

Then, since the purpose is to match a given ink-vector and NPacs vary in their ink-use, another constraint is needed. If the target ink-vector (ink-use) is $\boldsymbol{v}_i$ then the resulting NPac needs to match it and satisfy the equation shown above:

$$\boldsymbol{n}_i \cdot N = \boldsymbol{v}_i$$

Importantly, these conditions can be re-formulated as:

$$A \cdot \boldsymbol{n}_i \le \boldsymbol{b}$$

$$C \cdot \boldsymbol{n}_i = \boldsymbol{d}$$

Whereby [***A***, ***b***] define the inequality constraints imposed by convexity, while [***C***, ***d***] define the equality constraints, both that of area coverages adding up to 1 as well as the NPac resulting in a desired ink-vector. The general form of the above matrices can be written as follows:

$$A = \begin{bmatrix} I \\ -I \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{0} \end{bmatrix}$$

$$C = \begin{bmatrix} [1, 1, \ldots, 1] \\ N \end{bmatrix}$$

$$\boldsymbol{d} = \begin{bmatrix} 1 \\ \boldsymbol{v}_i \end{bmatrix}$$

where $\boldsymbol{I}$ is the *MxM* identity matrix, $\boldsymbol{0}$ and $\boldsymbol{1}$ are *Mx1* column vectors of *0*s and *1*s respectively. The domain over which NPacs that match a given ink-vector can be sought is defined by linear inequalities and equalities. Consequently, the solution space is a convex subspace and has a continuous volume of solutions resulting in infinitely many possibilities. It is over this domain that properties of NPacs can be considered.

In order to pick one of the possible NPacs, an optimization objective function is introduced – the mechanism through which particular print attributes will be controlled – starting with a simple example where NP area coverages are given weights or penalties. The objective function has the same dimensionality as the solution space, hence the number of NPs, *M=8* in this example. Given the earlier ordering of NPs, the simple objective function can be written as:

$$\boldsymbol{f} = [f_1, f_2, f_3, \ldots, f_M]$$

Whereby its meaning is one of weighing the contribution of each of the NPs: by weighting an NP_j with f_i it's relative value to all other f's determines the relative importance or penalty of this NP's use. A simple example here is that of minimizing blank media:

$$\boldsymbol{f} = [X, 0, 0, \ldots, 0]$$

Where the blank media NP (first) is given a weight of X>0 while all other NPs are given 0. Therefore, the values of the objective function will be higher for high amounts of blank media and lower for low amounts of blank media.

The optimization problem can now be formulated as follows, joining the constraints and the objective function:

$$A \cdot \boldsymbol{n}_i \le \boldsymbol{b}$$

$$C \cdot \boldsymbol{n}_i = \boldsymbol{d}$$

$$\min_{\boldsymbol{n}_i} \boldsymbol{f} \cdot \boldsymbol{n}_i + \boldsymbol{x}$$

Where $\boldsymbol{x}$ is a constant offset vector of the same size as $\boldsymbol{f}$. Since the constraints are convex and the objective function linear, this is known as a linear programming [5] problem that can be solved with known computational methods (such as MATLAB's or scipy.optimize's linprog). More generally however, the objective function can be written as:

$$\min_{\boldsymbol{n}_i} \frac{1}{2} \boldsymbol{n}_i^{T} H \boldsymbol{n}_i + \boldsymbol{f}\boldsymbol{n}_i + \boldsymbol{x}$$

For some ***H***, ***f*** and ***x***, whereby here the quadratic term ***H*** is introduced to be part of the minimization. If ***H*** is not an identity matrix then the above is referred to as quadratic programming [6] and, like its linear sibling, has well established methods for being solved using existing packages in MATLAB, scipy and others. In general linear and quadratic programming are examples of convex programming techniques [6], a sub-field of mathematical optimization.

The above framework therefore follows a sequence:

1. Given an existing NPac, compute its ink-vector.
2. Construct the linear inequalities and equalities as shown above, based on the set of NPs involved in the NPac, the target ink-vector and the target optimization function (as defined by *[**H**, **f**, **x**]*).
3. Compute new NPac that matches target ink-vector and is optimal in terms of the pre-determined objective.

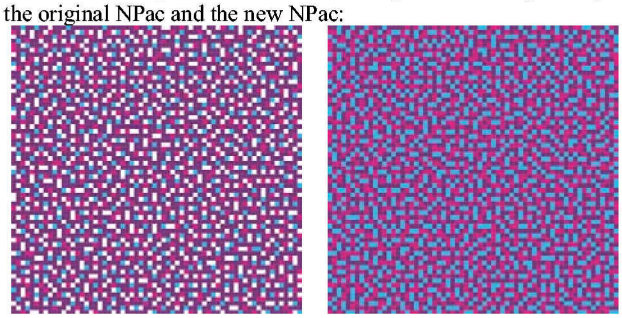
Following on from the simple example introduced at the beginning of this section, let $\boldsymbol{v}_i$ = $%& (%& (%&) be the input ink-vector and let $\boldsymbol{f} = [1, 0, 0, \ldots, 0]$ and $\boldsymbol{x}$ being zeros, be the objective function, over the same set of *M=8* NPs for a CMY ink-set and a binary deposition

[1] Percentages on a 0 to 100% range will be used for NPacs in spite of the equations assuming fractional values on a 0 to 1 range, while fractional values will be used for drops in ink-vectors. This is in order to clearly distinguish the two domains, such that 0.5 Cyan for an ink-vector is not confused with 50% of the single drop Cyan NP.

system. Computing the linear programming optimization yields the following NPac:

*"[# $% (\% (\% (% () % (% (% (%).

Note how *"[uses no white space at all, whereas the previous example shown earlier *"\ # $+% (, % (, % (% (-% (% (%). used 20%. To visualize this, below are two halftone patches corresponding to the original NPac and the new NPac:



! "#$%&'()'* + ,-./0&1'2+.'3/%%&14/05'./'.6 /'5"-&%&0.7 8+31 *"9&-.:'+05' *"[1%'#2.:'2+.'12+%&'2&'1+; &'"0<=>&3./%'!"?"0'/0&'3+1&'$1"0#'@AB 'C,+0<' 4'D&,1'9&-.:'+05'C/ 2'1"0# &'E F+0'+05'1"0# &'G +#&0.+'4'D&,1'+1'6 &,,'+1'2&' />&%4%"0.'-/%'E F+0'+05'G +#&0.+H'0'2&'/ 2&%3+1&'0/ 'C,+0<'4'D&,1'+%&'$1&5'"0' 2&'2+ ,-./0&H/0 F'E F+0HG +#&0.+'+05'2&'E G '/>&%4%"0.I'

The above example exposes an important consequence of NPac optimization. If a given NPac is mapped to an ink-vector and subsequently a new NPac is computed, that matches the original ink-vector, using some optimization criteria such that the original NPac and the new NPac don't match, then their resulting color does not need to match either. The magnitude of this difference will vary depending on how different the original vs new NPacs are but can be significant. This is important to keep in mind and has consequences in how this approach can be used. The figure below shows these differences based on a set of 854 NPacs that have been processed using a blank space maximizing optimization, with a mean difference of 0.7, 95th %tile of 1.6 and maximum of 4.3 ΔE2000.

Finally, the approach described here equally applies to cases where changes to the ink-vector or the set of NPs are desired. For example, calibrating an NPac, which amounts to applying a change in the ink-vectors, is performed as follows:

*"4/ # !"

!"- # 1! " ' a

Where *v* is a vector of calibration factors for each of the inks and ' the dot-product, such that a a # $%& (, &c (,) would represent a reduction by 10% for Cyan, an increase by 5% of Magenta and an unchanged Yellow. Once the new, target ink-vector !"- has been computed, the same process applies as shown before.

Likewise, if the set of NPs changes, for example if the drop sequence changes from [0, 1] to [0, 1, 2] and an additional set of NPs is available (e.g. not the full set possible with [0, 1, 2]), then the new matrix N is replaced with:

% , % % , , % , + % % + + % +

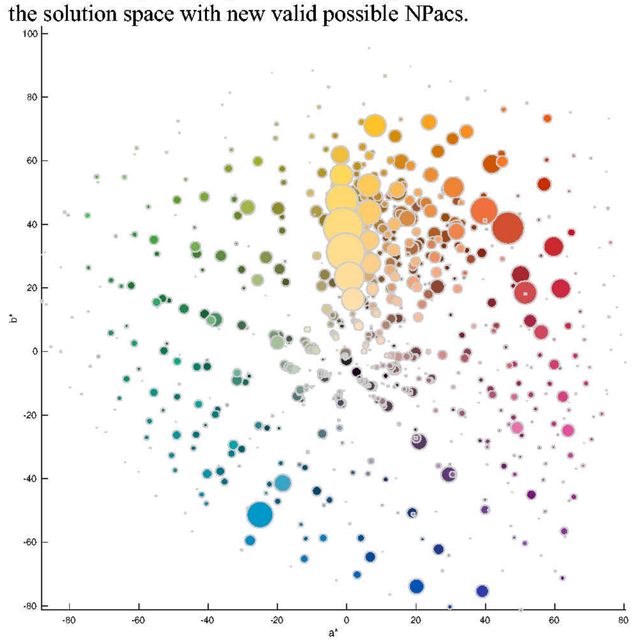
/ ^{0} # 12%1%1, 1%1, 1%1, 1, 1%1+1%1+1%1+1+3

% % % , % , , , % % + % + + +

And the solution vector *" then exists in reference to the new set of NPs, here using a simple white-space minimization objective function:

*"d # $% (\c (\c (% (e (% (% (% (% (e (e (% (- (% (% (%).

Which can be also written as [35% C, 35% M, 8% CM, 8% CC, 8% MM, 6% CCMM]. Note that no discrimination between 1 or 2 drop states or overprints took place here. However, it should be clear from the above simple example that extending the set of NPs increases the solution space with new valid possible NPacs.



! "#$%&'@)'E / ,,/%'5"-&%&03&1'C&,6 &&0'+'1&.'/-'7 8+31'+05'2&'%'/4."; 'U&5' 3/$0 &%4+%1'%$1"0#'+'1"; 4 &'6 2"&=14+3&'; +D"; 'U+."/0','0&+%'/C&3."5&' -$03."/0:I'

9 %6. $(&$%6" - $2 (, " 2' $612 (7- . ' $6, . # (

While the above examples serve to describe the optimization framework in general terms, the motivation behind its introduction is that of controlling specific print attributes via halftone optimization. In this section a number of examples of objective functions and their relation to such attributes are shown. In general, they share the common objective of maximizing some aspect of image quality, such as the absence of grain, robustness, line quality, etc. however other objectives can also be introduced and will be explained below.

The first and simplest example is that already mentioned in the previous section: minimizing the amount of blank pixels in printed output. The intuition behind such an objective is that the blank media NP has the highest L* with respect to all other NPs and no matter how well it is distributed spatially, its contrast with neighboring NPs will be high. Also, more blank pixels means less uniform coverage by ink in many cases. In order to achieve this, only the linear weights are needed, as shown before:

M # $Q (% (% (L (%)

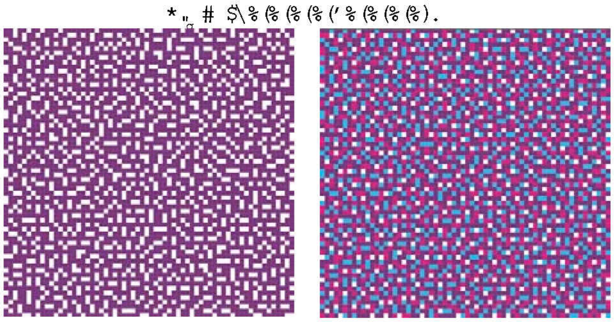
Where X is an arbitrary non-zero value. Figure 1 shows an example of this objective function on the right where it can clearly be seen that there are no blank pixels at all.

The next element to introduce is control over side-by-side or overprinting NPs. Maximizing overprinting, while at the same time minimizing white, will yield an NPac that has low NP contrast and will result in patterns that are, in general, lower grain, since grain relates to both the spatial distribution of the available NPs as well as their local color, and especially lightness, contrast. In this case the weights would be defined as follows:

M # $Q (f (f (f (% (% (% (%)

Where X and Y are, again, non-zero values acting as penalties to the corresponding NPs – the blank NP and the single-ink NPs in this case. Note that here the magnitude of X and Y is important and will affect the balance between penalizing blank media (X) and penalizing single-ink NPs (Y), in order to favor overprinting. The reason for this is that favoring overprinting and penalizing the blank NP can go against each other – i.e. the NPac with least white may be one that never overprints. The figure below shows the case where X7 Y (* "g) with X>Y being the same as Figure 1 right (* "f), such that:

* "g # $\% (% (% (% (' % (% (% (%) .



! '#$%& L)'MD+ ; 4 ,&1'/-'/ 2&%'/4 ."; 'U+ ."/0'/CK&3 .'>&'-$03 ."/01H'4&0+ ,'U'0# 'C,+0<' 14+3&'+05 '1 '0# ,&='0<'7 8 1 '9,&-.:H3/%%&14/05 '0#'./'* "d +05 '+ '5 "-&%&0 .'C+ ,+03&' C& ,6 &&0 '4&0+ ,'U'0# 'C,+0<'4 'D& ,1H/>&%4%'0 .'0#'+05 '1 '5&=CF=1 '5&'4%'0 .'0#H 3/%%&14/05 '0#'./'* "h I'

A symmetrical and alternate approach is to penalize overprinting and blank media, where the objective function is written as:

M # $Q (% (% (% (£ (£ (£ (i)

Note that here the combination of 2 (Y) or 3 (Z) inks may be penalized differently if other aspects (such as coalescence, mottle etc.) are relevant. Also, some of these weights may be redundant if X >> Y or Z since the lowest blank media option always will use the single-ink NPs as much as possible. The value of this objective function is especially for delivering uniformity of ink-use or coverage and also for details or lines where using as little blank space as possible is beneficial. An example here is:

* "h # $, % (+% (+% (% (c% (% (% (%) .

Finally, by adding a quadratic element, it is possible to include, for example, metrics relating to another NPac, such that the new NPac minimizes the M–dimensional Euclidean distance to the original NPac, while also introducing some penalization of NPs via the linear term. The quadratic term $\boldsymbol{H}$ to minimize Euclidean distance is defined as:

Y # 1* "k 0 4 * "k

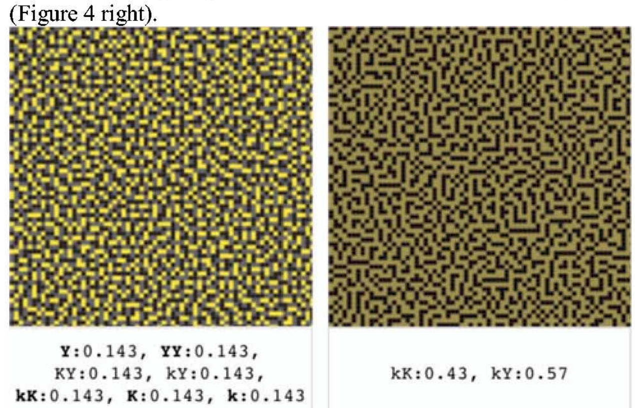
Such that a new NPac * "j will optimize both the distance to * "k as well as the linear term $\boldsymbol{f}$, is used. Yet another use of the quadratic term is to introduce a transformation of the NP domain into another one, for example based on the lightness difference among the NPs.

The above examples address specific halftone quality attributes by means of the convex optimization framework described earlier. Six NPacs have been shown, different from each other but matching in terms of the ink-vector and they will result in different print attributes, including their color, grain.

The sequence described in the previous section, coupled with the objective functions described here can be applied in a number of scenarios and a variety of applications, such as defining a target NPac metric and optimizing all nodes of a given LUT, using a series of target NPac metrics in order to explore different NPacs for the same ink-vector, given per-ink change factors (e.g. increase cyan, reduce magenta) use the above mechanism to introduce such changes in a LUT (i.e. calibration). The following section reports and discusses results of applying this framework to the examples mentioned.

; 2#-<$# (, 7(=&<7$, . 2 (, 4 $$5 6>&$6, . (

Figure 4 shows the process conceptually for a more complex example, using two black inks k and K as well as a Yellow, all of which can have [0, 1, 2] drops. At the input, the NPac uses seven NPs that have high contrast among them: one and two drops of Yellow and a variety of black-ink NPs. With the objective of maximizing overprinting (a proxy for minimizing contrast), maintaining the ink-vector (no change in the amounts of ink) and with the NPac constraints of convexity, a new NPac is computed that only uses two NPs, both overprinting inks, in one case black and yellow, resulting in an NPac that has lower contrast and therefore lower grain, as can be seen even in the halftone preview (Figure 4 right).

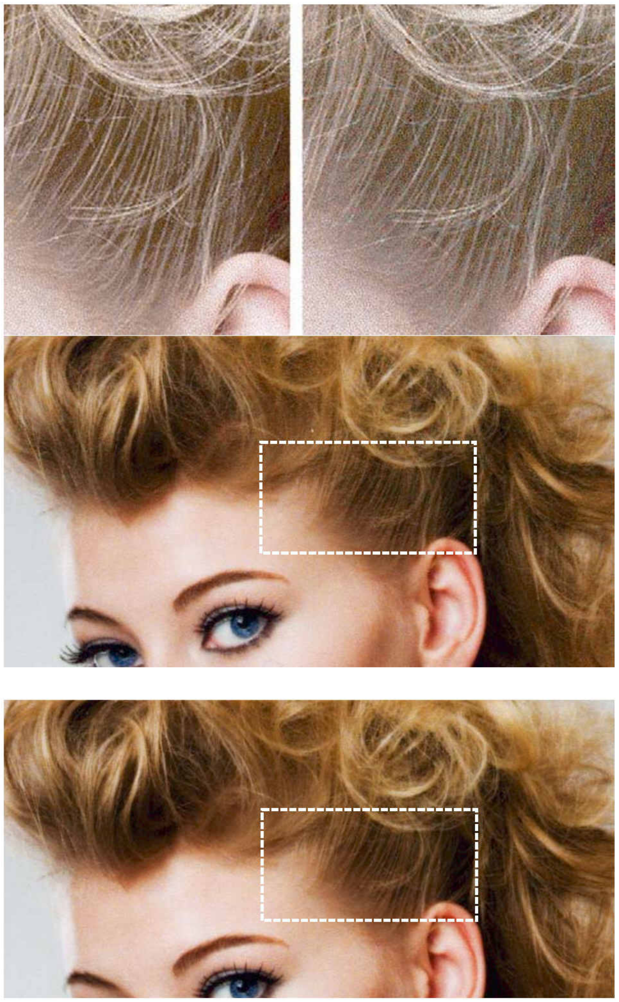


! '#I'N)'O0 "'04$.'7 8 +3'9+05 '2+ ,-./0&:'12/6 0'/0'2&'&-H+05'2&'/$ .4$.'7 8 +3' $1 '0#'+0'/>&%4%'0 .'0#'; +D"; 'U+ ."/0'+05'2&'%&1$,.'0# '2+ ,-./0&'9%'#2 .:I'

In the following sub-sections, each of the applications is briefly described and results of applying the convex programming halftone optimization approach to complex printing systems with 9 to 12 inks and a variety of drop sequences are shown in the accompanying Figures.

GHI %'J @#)7)83#)' . %

The first example here is that of post-processing an existing HANS LUT. The purpose of this is to compensate for re-sampling of the LUT, e.g. if it was built for a 9^3 case and it is desirable to have a 25^3 version whereby the original was built with, e.g. white-minimization and overprinting in mind, a direct up-sampling and re-interpolation will result in convex combinations, which means no new NPs will be introduced even if they are more advantageous in terms of grain. Hence, the optimization can be performed after the up-sampled LUT is computed on a node by node basis, in each case making the best possible use of all available NPs for the given node. Figure 5 shows such an example from the HP Z3200 12 ink photo printer with a [0, 1, 2] drop sequence, where a simple version using only 31 NPs (the minimal set of single-ink NPs) can be compared against one that also uses NPs that combine (overprint) NPs, with a clear advantage in terms of grain and IQ.



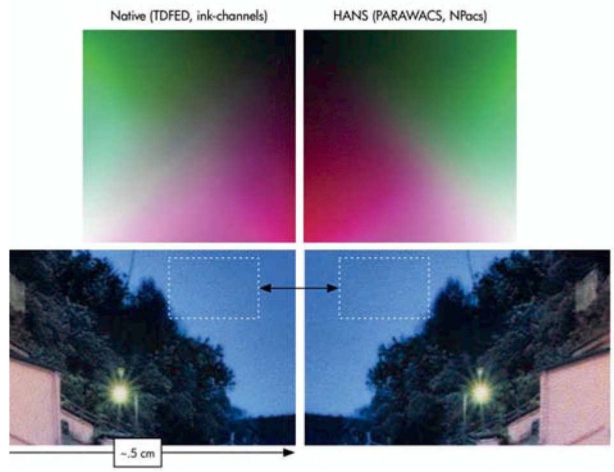
! '#'P)'Q 4 ."; 'U+ ."/0'&D+ ; 4 ,&)'2+ ,-./0&1'9./4',&-.'+05'%'#2 .:'+05'13+01'/-'4%'0 1' 9; '55,&'+05'C/ ../ ; :'-%/ ; '+'4'4&,'0&'$1'0# 'L('7 8 1'>'+ '7 8 '%&5$3."/0'+05'1'5&= CF=1'5&'/4 ."; 'U+ ."/0'9./4'%'#2 .'2+ ,-./0&'R'; '55,&'13+0 :'+05'/>&%4%'0 .'0#'+05' 6 2"&'/4 ."; 'U+ ."/0'9./4',&-.'2+ ,-./0&'R'C/ ../ ; '13+0 :I'* '#2 ,'#2 .&5'+%&+1"0'.2&' 13+01'12/6 '4+%.'3$,+%F'0/ .'3&+C,&'5"-&%&03&1'+05'3/%%&14/05'./ '%&#"/01' 12/6 0"0'.2&'2+ ,-./0&'3%/41I'

A'7 @' $#).2%"K)4#).2%@)@"F). "4%

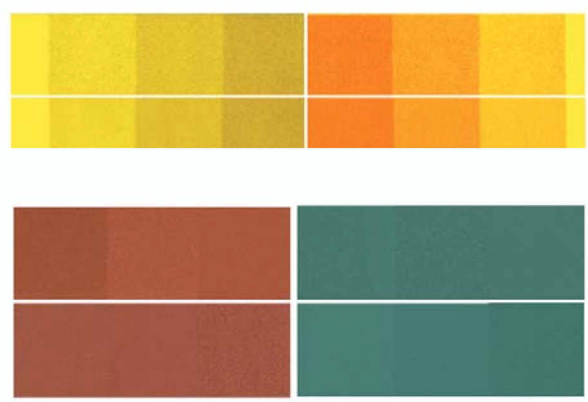
Likewise, NPac optimization can be used for importing existing LUTs that are not HANS based by post-processing them. This process has two main proposes: to simplify the NPac structure (the complexity of the NPacs or the number of NPs per NPac) and to mix the inks with dot-on-dot overprinting as much as possible. These two objectives are aimed at improving the input to PARAWACS halftoning, which benefits from fewer NPs while the overprinting optimization instead is akin to the effect of error diffusion which implicitly also aims to overprint to have smaller color or lightness differences between the halftoned NPs. The result is an improvement in the grain of the printed plots as can be seen in Figure 6 using the example of a HP Z3200 12-ink photo printer, printing on glossy substrate.

6 $3). %937 @F). 2%

Some of the computational metrics act as proxies for perceptual ones, such as the NP contrast minimization (or its simpler form, white space minimization) acting as a means to vary grain. By modifying the objective function, it is possible to build versions of NPacs with varying levels of the metric and thus build sets of NPacs that explore grain variety. Figure 7 shows examples of this for a nu



! '#$%&'S)'T ; // .20&11'3/ ; 4+%'1/0'9./4'4+'%H+44%/D"; + .&,F'+ .'(AAB 'J// ; :' +05'#%+'0'9C/ ../ ; '4+'%H6 '5 .2'3%/4'3/%%&14/05'0#'./ 'UAIP'3 ; :'/-'.2&'0+ .'>&' 5/ ; +'0'9,&-.'3/ ,$; 0 :'>1'* O7 T '%'#2 .'3/ ,$; 0 :I'V0'.2&'./4'4+'%'.2&' ; +'0' 5"-&%&03&1'3+0'C&'1&&0'+%/$05'.2&'0&$.%+ ,'+D"1H6 "2'0+ .'>&'WX ! MX '2+>'0#' '0>&%1"/01H6 2"&'"0'.2&'C/ ../ ; '4+'%'"; 4%/>&5'#%+'0'Y$+ ,"F'3+0'C&'1&&0'9/>&%' +0'+ ,%&+5F'2'#2 'VZ '4%'0 .:I'



(a

(t

! '

; ·

; ·

L ! 30%937 @F). 2%

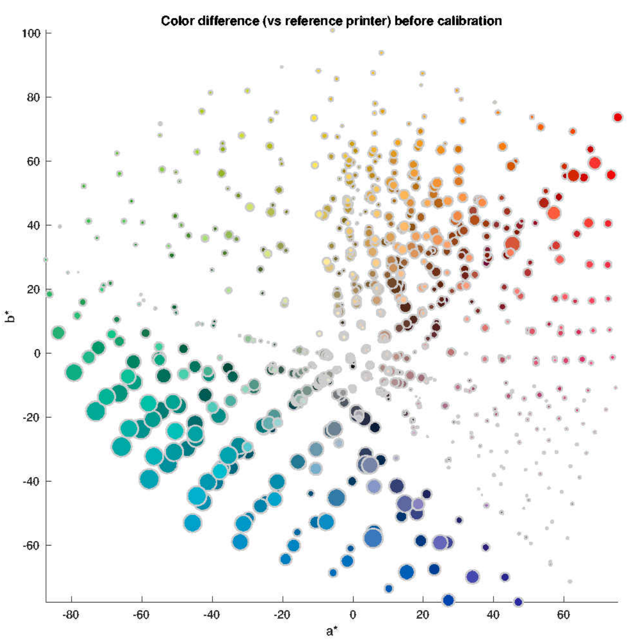
In the process of exploring the vast domain of NPacs (e.g. there are $3^{12} = 531{,}441$ NPs for a 12 ink system with [0, 1, 2] drops) it can be valuable to explore ink-vector neighborhoods (the same system has 12 inks, so a 12D domain of ink-vectors vs a 3^{12}D domain of NPs) first, followed by exploring the NPac variety for a constant ink-vector. Given a sampling of ink-vectors coupled with a variety of objective functions, a set of locally varying NPacs, anchored at the ink-vectors can be explored in the described way. Thus, both classes of equivalence are generated in terms of constant ink-vectors, as well as classes of equivalence in terms of the NPac properties optimized (all NPacs that minimize white, all NPacs that maximize overprinting, etc.).

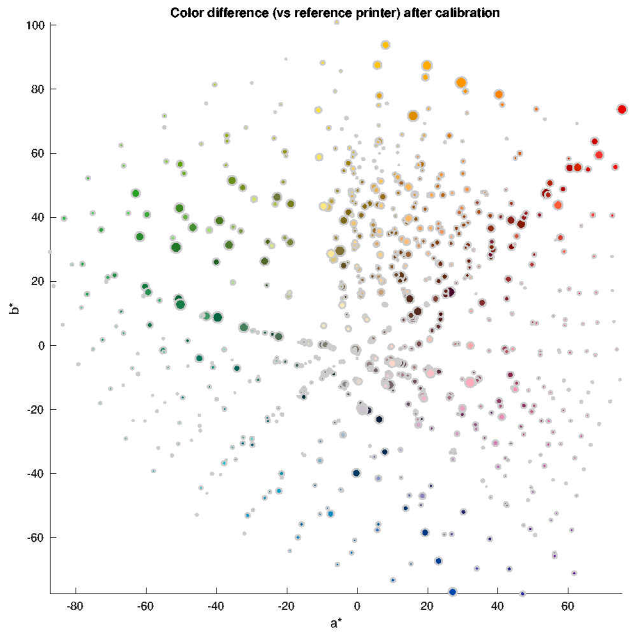
C ' F $%C3F< $3#)' . %

Finally, given a LUT and calibration values in ink-channel terms – i.e. factors of change for each of the inks – a new LUT can be computed by modifying each node according to the ink-vector change required to compensate, while also maintaining the objective used to build the original LUT as described above. An important property of this approach is both that image quality considerations are part of the calibration process as well as it handling well both increases and decreases in ink-amounts. Figure 8 shows results of applying calibration to an existing system and its performance after calibration vs a reference. The color difference before calibration (current printer vs a reference) was a median of 1.2, 95th %tile of 2.7 and a maximum of 3.5 ΔE2000, while after calibration the difference vs the reference was a median of 0.7, 95th %tile of 1.4 and a maximum of 1.9 ΔE2000.

0 , . ' <-#6, . # (

The overall result of an imaging pipeline depends both on the spatial structure of the color–forming patterns as well on the atomic components that these patterns are made of. In this paper a novel approach to controlling the at-pixel states was described that affords great control over the type of pixels that are desired. This enables careful tuning of color look-up tables, generating samples that vary in secondary attributes while keeping an ink-vector constant and even color calibration. Importantly, all of this processing can be done with explicitly defined metrics that affect the overall outcome, such as minimizing white-space, maximizing or minimizing overprinting, minimizing distances to other NPacs etc. The approach described here is at the heart of the resource building process for the highest IQ HP printing product line, the HP DesignJet Z6 and Z9+ graphics and photo printers.





! '#I'])'E / ,/'%5 "-&%03& '5 '1 .%C$.'/ 0 'C&-/ '%& '3+ ,'C%+ .'/ 0 '9 ,/ 4 :'+ 0 5 '+ - .&%3+ ,'C%+ .'/ 0 ' %C/ . ./ ; :'$1'0#'+'6 2"&=; '0"; '0'0#H/ >&%4%0 .'0#'; +D"; '0'0#'3+ ,'C%+ .'/ 0 I'

; 2 7 2 %2 . ' 2 # (

[1] Morovič J., Morovič P., Arnabat J. (2011) "HANS – Controlling Inkjet Print Attributes Via Neugebauer Primary Area Coverages," *IEEE Transactions on Image Processing*, 21/2:688–696.

[2] Schramm M. T., Gondek J. (1999) "Tone dependent plane dependent error diffusion halftoning" USPTO US6501564B1

[3] Li P., Allebach J. P., (2005) Tone-Dependent Error Diffusion,

[4] Morovič P., Morovič J., Gondek J., Ulichney R. (2017) Direct Pattern Control Halftoning of Neugebauer Primaries, IEEE Transactions on Image Processing, 26/9:4404-4413.

[5] Chvátal V., (1983) Linear Programming, W. H. Freeman & Co Ltd

[6] Boyd, S. P., Vandenberghe, L., (2004) Convex Optimization, Cambridge University Press

! - $= , %() 6, + %&4 = ? (

Peter Morovič received his Ph.D. in computer science from the University of East Anglia (UK) in 2002 and holds a B.Sc. in theoretical computer science from Comenius University (Slovakia). He has been a senior color and imaging scientist at HP Inc. since 2007, has published 50+ scientific articles and has 100+ US patents filed (33 granted) to date. His interests include 2D/3D image processing, color vision, computational photography, computational geometry. His Erdős number is 4.

Ján Morovič received his Ph.D. in color science from the University of Derby (UK) in 1998, where he then worked as a lecturer. Since 2003 he has been at Hewlett-Packard in Barcelona as a senior color scientist and later master technologist. He has also served as the director of CIE Division 8 on Image Technology and Wiley and Sons have published his 'Color Gamut Mapping' book. He is the author of over 100 papers and has filed 100+ US patents (36 granted).

Víctor Diego was born in Entrambasaguas (Spain). He graduated in mathematics in 2012 at the University of Cantabria, received his master's degree in advanced mathematics and mathematical engineering at the Polytechnic University of Catalunya in 2013 in Barcelona and his Ph.D. in discrete mathematics and graph theory in 2017 at the same university. Since 2017 he has been an R&D developer at HP in Sant Cugat, Barcelona.

Javier Maestro has a MSc in communications engineering from the Centro Politécnico Superior in Zaragoza. Interested in color and image processing, he has worked for 7 years at HP developing imaging and printing pipelines for graphics printers, particularly focusing on the mapping of RGB spaces to HANS-based representations in systems with a high number of colorants.

Hector Gomez received his B.Sc. degree in Mathematics from the University of Barcelona in 1998. Since 2007 has been working in HP as a Writing Systems Engineer dealing with Image Quality problems, Servicing and Color calibration algorithms. His interests include Data Science, Machine Learning and Image Processing.

Sergio Etchebehere recieved his MSc in color science from University of Eastern Finland and Universidad de Granada in 2016, where then he later worked as a researcher. Currently he is a Color and Imaging Scientist at HP Inc. His research interests are color science, image processing, human vision and computer vision

Xavier Fariña was born in Barcelona in 1976. He holds a master's degree in mathematics and a master's in big data and business intelligence. Before joining HP in 2011 he built his professional career in several companies related to software development, performing roles like Analyst, DBA, Web and mobile developer or UX designer. He loves playing board games, reading and spending time with his wife, his two little children and his two cats.

Pere Gasparin holds a bachelor's degree in Indrustrial Engineering in Electronics from the Universitat Politecnica de Catalunya, and a bachelor's degree in computer science from the Universitat Oberta de Catalunya. He has been working at HP in Barcelona since 2002, involved in Writing Systems and Color and Imaging teams for the development of large format inkjet printers.