https://doi.org/10.2352/issn.2169-2629.2020.28.48

Revisiting print-attribute optimization: a direct pattern generation approach

Peter Morovic, Hector Gómez, Ján Morovic, Pere Gasparin, Tanausú Ramírez, Xavier Fariña, Sergio Etchebehere,
HP Inc., Sant Cugat del Valles, Spain

Abstract

The properties of prints are not fully determined by the materials they are composed of and the method that was used to compose them. These merely set limits to what a print's properties, such as its colors, sharpness, smoothness, color inconstancy and level of ink use, will be and it is the role of a printing system's imaging pipeline to select a particular combination. Conventionally such choices are implicit in how resources for a pipeline are built and can be improved with experience and trial an error. Performance can be improved though by optimizing for specific attributes, as was previously shown for color consistency, ink use and grain among others. A key constraint that remains here is that optimization is performed on the basis of sampling and search strategies, which have inherent limitations. This paper presents a direct, analytical approach to optimization that hinges on the insight that it can be performed in a convex space even when the properties involved in the optimization do not relate to each other in a convex way. The result both improves performance versus previous methods and does so in considerably less time.

Introduction

With ever more advanced printing systems that can use a variety of inks (e.g. colors, varnishes, pre-treatments, overcoats) and deposition mechanisms (e.g. inks jetted with multiple or variable drop volumes), the ability to computationally optimize print attributes becomes ever more important.

Essentially, print-attribute optimization amounts to choosing patterns (e.g. ink-vectors, Neugebauer Primary area coverage (NPac) vectors, halftone patterns) that are particularly good in terms of some of their printed characteristics. Key to this challenge is understanding the attributes in the first place: how does ink-use, grain, robustness, color inconstancy, coalescence etc. relate to a pattern that can be generated as an input to a printing pipeline and result in printing instructions that in turn yield the optimal printed output?

Furthermore, many decades of artisanal intuitions are deeply engrained in how attributes such as ink use or grain are thought of, and while they can and do lead to good results in many cases, new imaging pipelines open up possibilities that render past intuitions unreliable.

While some print attributes are easier to model than others, thus far the approach to optimization has been broadly similar and independent of the metric being optimized: search the solution space (using a variety of techniques including RANSAC [11], Monte Carlo based approaches or gradient descent) and pick the best patterns in a local color neighborhood. Doing this over the entire addressable gamut of a printing system then results in a sufficient sampling of optimal patterns and imaging pipeline resources, e.g., color separation look-up tables, can be obtained via interpolation.

This family of approaches has been applied to optimizing the color gamut and color inconstancy of prints [12] and later, in the context of the HANS imaging pipeline, to ink use [3,4] and grain [13]. However, there are several limitations to such approaches: first, that they are computationally expensive (potentially 10s or 100s of millions of patterns need to be evaluated to robustly pick optima, leading to long computation times), second, that whether the choice is strictly optimal is uncertain and third, that by virtue of the vastness of the solution space, constraints need to be imposed for the computation to be tractable.

A 4-ink system with 4 states (no ink, 1, 2, 3 drops of ink) results in 256 possible Neugebauer Primaries. To check all possible ways to express a given colorimetry within such a printing system's domain, at least all possible tetrahedra need to be built, of which there are 175 million in this case. Checking for inclusion and evaluating a metric for, say 1000 colorimetries covering the color gamut, each against 175 million possible tetrahedra is not feasible in reasonable time. Even if it were, this would only cover the simplest NPacs, involving 4 NPs, while the optimum may involve more than 4 out of the possible 256 NPs being combined to render a color optimally. Furthermore, a 4–ink system is unusually simple and it can easily be seen how such approaches quickly becomes too expensive and necessitate imposing constraints that may compromise optimality.

Convex methods for non–convex color formation

When looking at a print, the color of a pattern of colorants and colorant combinations distributed over an area that is below the human visual system's spatial acuity threshold is characterized by the Neugebauer equation [1]. Here, a pattern C, formed by combining some of a printing system's Neugebauer Primaries (NPs), can be characterized by its NP area coverage (NPac) vector – $NPac_C$ as follows:

$$T(NPac_C) = \sum_{i=1}^{k^n}(w_{Ci} * T(NP_i)) \quad (1)$$

where k is the number of colorant levels per colorant per unit area, n is the number of colorants (i.e., the number of possible colorant combinations per unit area being k^n), $\sum_{i=1}^{k^n} w_{Ci} = 1$ (i.e., the weights are convex), NP_i is the *i*-the NP, and *T()* is color (e.g., CIEXYZ, or Yule–Nielsen–corrected [2] CIEXYZ taking into account non–linearity from lateral scattering). In other words, knowing the colorimetries of the components of a sub-spatial-acuity-threshold pattern lets us predict the color of that pattern, as integrated by the human visual system, simply by computing their area-weighted combination, which is convex when the weighting is made relative to a given area and therefore sums to one.

This, however, is easier said than done, and the challenge lies in the complexity of an actual printed pattern. Taking a simple, 3-colorant, binary example (e.g., the one shown in Fig. 1., where red, yellow and gray inks are used with up to one drop of ink specified for each location), it can be seen that even here there is an explosion of combinations, far beyond what a naïve application of the Neugebauer equation would assume.

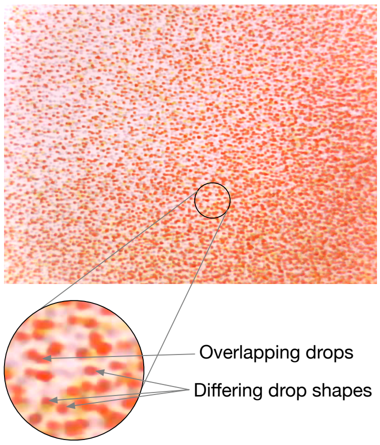

***Figure 1.** Scan of a 3-colorant print using red, yellow and gray ink, with a zoomed-in region showing complex ink drop combinations.*

Instead of seeing only 2^3=8 NPs expected in this case (blank substrate, R, Y, G, RY, RG, YR and RYG), the superimposition of drops of ink that vary in size and shape and whose placement on the substrate is influenced by other drops of ink and varies from drop to drop, results in a far greater number of Neugebauer Primaries in the physical print. Instead of there being only two levels, there are in fact at least 5 such levels just due to the fact that near-circular shapes are used to fully cover an area. The example in Fig. 2 shows the combinations of drop layers corresponding to a single drop being specified for each of a print-resolution halftone's pixels. Instead of the NPac [C1=100%] specified in the digital halftone (i.e., one drop of cyan ink covering all of some area), resulting in a single, homogeneous layer of ink, the resulting print would contain the following NPac [C1=42%, C2=44%, C3= 9%, C4=5%] for the area shown in Fig. 2. Here the digital halftone's NPac is best thought of as a token for the resulting, printed NPac.

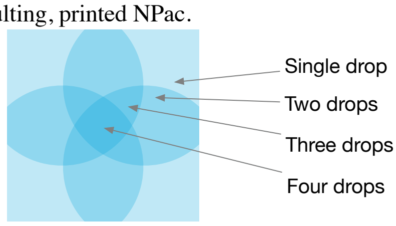

***Figure 2.** Ink layers corresponding from single drops specified per print-resolution halftone pixel.*

This alone would lead to not 8 but 5^3=125 NPs even in the simple, 3–ink case. Adding variation in placement to the mix can further increase this number and, in more heavily inked parts of a print the number of levels per ink at any one location can go to 6 (→ 216 NPs) or more. For a more realistic, but still simple, example of a CMYKcm ink set where up to 3 drops per ink per pixel can be specified, the result would be not 4^6=4,096 but 13^6=4,826,809 NPs, simply due to no-tessellating drops. The first challenge here is that it is over this domain that the Neugebauer equation postulates convexity. I.e., given accurate colorimetries of these NPs and the area coverages corresponding to them in a physical print, convex combinations should lead to accurate integrated color. This, by itself makes it impractical to operate in the domain of NPs in the physical print, which will be referred to as "effective" NPs (eNPs).

However, there is a yet more severe obstacle at play here. Even if it were possible to know both the colorimetries and relative area coverages of eNPs, these don't constitute a print control domain. Unlike the NPs of digital halftone patterns, which will be referred to as "synthetic" NPs (sNPs), not all eNPs are physically realizable. E.g., the eNP [C1=100%] is impossible in a printing system that deposits colorants resulting in non-tessellating shapes on the substrate. Attempting to control a printing system with eNPs (where convexity is postulated to hold) would require solving two difficult problems: first, determining whether an eNPac (the analogous eNP area coverages) is physically realizable and, second, identifying the sNPac (the analogous sNP area coverages, equivalent in previous literature to simply NPacs) that is a token for it and that would correspond to the desired eNPac when printed on the given system. These problems would render the convenience of the eNP domain's convexity irrelevant since they would require non–convex and costly computation.

Continuing to operate in the sNP domain, as has been the case in previous applications of HANS [3-7], then maintains the challenge of making choices in a space (that of sNPacs), which is not convex with regard to corresponding color. I.e., the convex combination of sNPs has a color that does not match the convex combination of its colorimetries. Instead, a different printed color results that can, however, be successfully predicted from an sNPac using a non–linear, spectral extension of the Neugebauer and Yule–Nielsen models [8]. The result is a way to operate in the sNPac control domain and to have accurate color for each sNPac, but at the cost of non–linearity and also non–convexity. Convexity does not hold because the non-linear model does not satisfy the requirement for preserving convexity, which is for the mapping to be a collineation [14] (i.e., a mapping such that collinear points remain collinear after being mapped). The reason for this, in turn is twofold: both a non-linearity between the convex color prediction from sNPacs and actual ink–ink–substrate interactions and the fact that in the eNPac domain new eNPs enter in a way that does not correspond to sNP changes in the corresponding sNPac.

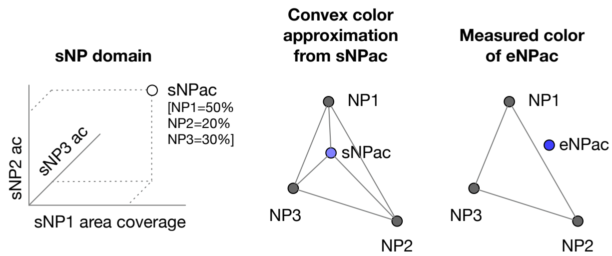

***Figure 3.** (Left) an sNPac in the sNP domain, (center) a convex color prediction made based on sNP colorimetries with sNPac area coverages as convex weights, (right) the measured color of the printed eNPac for the halftone of the sNPac that was a token for it.*

There is, however, a way to maintain the use of the sNPac domain while applying optimization that relies on convexity, which presents itself when the basic requirements of optimization are revisited. Namely, that it be a way to obtain a set of optimized NPacs that span the available color gamut and that these NPacs be well distributed over that volume. Taken strictly, it seems like there is no way to reconcile the use of the sNPac domain while obtaining optimal choices for specific colorimetries across a color gamut. However, the requirement for optimization to yield answers for specific colors is unnecessary, so long as a set of optimizations spans and covers the addressable color gamut well. It is sufficient for the full gamut to be addressed, even if with some irregularity of sampling. This in turn can be achieved by applying the Neugebauer equation to sNPacs, which will yield inaccurate predictions, but ones that are obtained in a convex way and therefore enable the application of convex optimization techniques. The end result here is still a set of optimized sNPacs that span and sample the available gamut, but not at the colors that the direct application of the Neugebauer model suggests. Instead, the optimized NPacs correspond to other colors that can be computed for them, after convex optimization, using non-convex methods or even printing and color measuring. Fig. 3 illustrates the process on a simple 3-sNP example and the following sections will present the approach formally and then report results of testing its assumptions.

Optimal Pattern Generation

Let us assume a printing system with *I* inks and a set of *N* NPs, and let us further denote them in matrix form as an *NxI* matrix $\boldsymbol{P}$ (as in primaries), such that for example one drop of the first ink will be [1 0 0 … 0], one drop of the second ink [0 1 0 … 0] and one drop of the first and second ink overprinting [1 1 0 … 0]. Furthermore, let us assume that the colorimetry of each NP is known as well (for a way to obtained NP colorimetries see [8]), denoted as the Nx3 matrix $\boldsymbol{P_C}$. Thus $\boldsymbol{P}$ and $\boldsymbol{P_C}$ are corresponding matrices, the first one expressing the NP (e.g. [0 0 0 1] would represent 1 drop of Black ink in a CMYK system), while the second one its corresponding colorimetry. Then, assuming the simple convex color formation model – suspending for a moment the knowledge that this does not hold true in general as shown in the previous section – if a convex *Nx1* vector $\boldsymbol{w}$ exists for a colorimetry $\boldsymbol{a_C}$ such that it can be written as:

$$\boldsymbol{a_C} = \sum_{i=1}^{N} \boldsymbol{w^i} * \boldsymbol{P_C^i} \tag{2}$$

Note that Eq. (2) above is a simplified version of Eq. (1) earlier, without the color mapping function *T()*. Then, using the same weights w_i, the corresponding NPac $\boldsymbol{M(a_C)}$ – again assuming this simplified convex color formation model – can be written as:

$$\boldsymbol{M(a_C)} = \begin{bmatrix} \boldsymbol{w^1} & \boldsymbol{P^1} \\ \dots & \dots \\ \boldsymbol{w^N} & \boldsymbol{P^N} \end{bmatrix} \tag{3}$$

Equation (2) and (3) don't yet express anything new, they simply mean that any colorimetry that can be written as a convex sum of the NP colorimetries, has a trivial valid NPac in this domain. If color formation in print was perfectly linear and convex then any random weight vector w, when associated with a set of NPs in $\boldsymbol{P}$ would result in exactly the colorimetry according to Eq. (2) above.

Let us continue assuming that we can rely on this model. In the context of optimization, we seek to answer the question: what is the set of all colorimetries for which there exist valid NPacs over some set of NPs? It is over this set that any optimization can take place. The answer is very simple and follows from the above equations, it is the set of all possible convex weights $\boldsymbol{w}$ applied to the NP colorimetries $\boldsymbol{P_C}$, which in turn is the convex hull of the NP colorimetries. Hence, the set of all colors that a system with $\boldsymbol{P}$ NPs is capable of, is the convex hull of the NP colorimetires $\boldsymbol{P_C}$. We can express this color volume V for a set of NP colorimetries $\boldsymbol{P_C}$ simply as follows:

$$\boldsymbol{V(P_C)} = \begin{matrix} \{\boldsymbol{w_j} * \boldsymbol{P_C}\} \\ \forall \boldsymbol{w}: \\ \boldsymbol{0} \le \boldsymbol{w^i} \le \boldsymbol{1} \\ \sum \boldsymbol{w^i} = \boldsymbol{1} \end{matrix} \tag{4}$$

This means that searching through the available color volume ***V(P_C)*** and expressing any valid colorimetry $\boldsymbol{a_C}$ as an NPac is a straightforward process and is, in fact, at the heart of previous optimization approaches [3,4].

However, turning the problem around and searching for all possibile NPacs that match a given color – i.e. computing metamer sets – is not as straightforward. Consider the following problem: what is the set of valid NPacs over some set of NPs that match a given colorimetry? The set of all possible NPacs $\boldsymbol{M^i}$ for a set of NPs $\boldsymbol{P}$, is the set of all possible convex weights $\boldsymbol{w}$ over the set of NPs, hence it's convex hull – analogous to equation (4) however in NP space rather than in colorimetry:

$$\boldsymbol{V(P)} = \begin{matrix} \{\boldsymbol{w_j} * \boldsymbol{P}\} \\ \forall \boldsymbol{w}: \\ \boldsymbol{0} \le \boldsymbol{w^i} \le \boldsymbol{1} \\ \sum \boldsymbol{w^i} = \boldsymbol{1} \end{matrix} \tag{5}$$

Note that both Eq. (4) and (5) are the same volume of NPacs and the volume of corresponding colorimetries are both fully determined by the set of weights $\boldsymbol{w}$. Hence the above inequalities applied to $\boldsymbol{w}$ delimit a convex polytope in *N* dimensions (albeit a simple one – effectively a hypercube).

Convex combinations of NPs weighed by their relative area coverages resulting in NPacs will use some amount of each of the available inks used by the printing system. This amount typically needs to have limits imposed, such as for the total amount of ink that a given substrate can absorb without causing artifacts and in some cases even constraints for each of the inks individually. Hence the next set of constraints is:

$$\sum_{i=1}^{N} \boldsymbol{w^i} * \boldsymbol{P^i} \le \boldsymbol{l} \tag{6}$$

Where $\boldsymbol{l}$ is an Ix1 vector of per-colorant ink limits, and:

$$\sum_{k=1}^{I} \sum_{i=1}^{N} \boldsymbol{w^i} * \boldsymbol{P^i} \le \boldsymbol{l_g} \tag{7}$$

Where the above sum is over all I inks and $\boldsymbol{l_g}$ a scalar denoting the total amount of ink not to exceed. Hence solving for the hafplane intersection of inequalities (5) to (7) results in the convex polytope of all valid NPacs for a given set of NPs that fall within the ink-limit of a given system.

Next, to find all NPacs that match a single given colorimetry, an equality constraint needs to be added to the inequalities above:

$$\sum_{i=1}^{N} \boldsymbol{w^i} * \boldsymbol{P_C^i} = \boldsymbol{a_C} \tag{8}$$

Since $\boldsymbol{a_c}$ is a 3-vector of colorimetries (e.g. XYZs or their Yule-Nielsen modified counterparts) Eq. (8) results in 3 equality constraints. Since we are dealing with a system that has noise it can be convenient to relax Eq. (8) into two sets of inequalities that delimit a cube in colorimetry, allowing therefore for a range of values (a local neighborhood), such that:

$$\sum_{i=1}^{N} \boldsymbol{w^i} * \boldsymbol{P_C^i} \le \boldsymbol{a_C} + \boldsymbol{tol} \tag{9}$$

$$\sum_{i=1}^{N} \boldsymbol{w^i} * \boldsymbol{P_C^i} \ge \boldsymbol{a_C} - \boldsymbol{tol} \tag{10}$$

Where ***tol*** is the tolerance (a scalar or a 3-vector) within which the target colorimetry should fall (e.g. for ***tol***=0.1, aC can be within +/- 0.1 in each of the three dimensions). For ***tol***=0 Eq. (9) and (10) are equivalent to Eq. (8).

To summarise, solving for all NPacs that match a given colorimetry (or are in its close vicinity), a halfplane intersection or convex hull problem – composed of the inequalities in Eq. (4) – (10) (with (8) and (9) with (10) being interchangeable), that define constraints on $\boldsymbol{w}$ – needs to be solved and since $\boldsymbol{w}$ is an *N*-vector (*N* being the number of NPs), the problem to solve is N-dimensional.

While solving for a convex hull in 3D is straight-forward and computationally cheap (MATLAB's built in convhull, python scipy's built in ConvexHull etc. can handle this with ease), doing so in higher dimensions can become prohibitive. Even for a simple system of CMY inks and a binary deposition scheme (no ink, some ink) there are 8 NPs, hence the problem would be an 8D convex hull computation. However, the space of NPs is typically much higher-dimensional and grows exponentially with the number of inks. A more realistic example system therefore is one that uses e.g. 6 inks (CMYKcm) and has the ability to deposit 4 states (0, 1, 2, 3 drops per pixel) resulting in 4096 valid NPs. While not all of these dimensions may be needed, the dimensionality will clearly be beyond the capabilities of convex hull computation possible today.

However, while computing all NPacs that match a given colorimetry is valuable and interesting, often it is NPacs of particular properties that are sought. An important insight here is that if these properties can be expressed as well-behaved functions – e.g. linear or quadratic – efficient ways to optimize over the convex hull are possible, without the need to actually compute the hull itself. The general framework of optimization to achieve this is defined as follows:

$$minimize: f_0(x)$$

$$subject\ to: f_i(x) \leq b_i,\ i = 1, \ldots, m \quad (11)$$

Where $f_0(x)$ is the objective function and $[f_i(x), b_i]$ the constraint functions. If both f_0 and all f_i are convex then Eq. (11) describes a convex optimization [9] and if furthermore $f_0(x)$ is linear or quadratic, then it describes linear or quadratic programming respectively [10].

Since all constraints introduced earlier can be formulated as convex (linear inequalities), optimal NPacs can be solved for so long as the objective function is convex as well. One of the simplest examples here is that of ink-use. Let $\boldsymbol{d}$ be an *Ix1* vector of drop weights, e.g. the weight or volume of a single drop of each of the inks, then the amount of ink used by a given NP is simply:

$$\sum_{k=1}^{I} \left(\sum_{i=1}^{N} \boldsymbol{w}^{i} * \boldsymbol{P}^{i} \right)_{k} * \boldsymbol{d}_{k} \quad (12)$$

Which can be simplified further by pre-computing the ink-use for each NP, denoted as the N-vector $\boldsymbol{m}_i$:

$$\boldsymbol{m}_i = \sum_{k=1}^{I} \boldsymbol{P}_k^{i} * \boldsymbol{d}_k \quad (13)$$

Hence if the objective function in Eq. (11) is defined as:

$$f_0(\boldsymbol{w}) = \boldsymbol{w} * \boldsymbol{m} \quad (14)$$

Which is a simple weighing function that results in penalizing the use of ink-use heavy NPs while rewarding the use of low ink-use ones – directly proportional to the amount of ink used by each of the NPs. Thus minimizing $f_0(\boldsymbol{w})$ results in NPacs that use the least amount of ink in total.

The above optimization can be solved for several target colorimetries, resulting in a representative set of optimal NPacs. However, as mentioned already, at the heart of this optimization is an assumption that is not true in general: that convex combinations of NPs result in a colorimetry that is equal to the same convex combination of the NPs colorimetries. Hence the next step is to correct for this by applying a non-linear color model as presented previously [8].

The color model used here has as its inputs NPacs and predicts reflectance and colorimetry as its output using supervised machine learning, which has been shown to work to high degrees of accuracy in this context. Starting with a training set of NPacs and measured reflectances, a second order polynomial with neighboring wavelength cross-terms is used to fit the training data and the results section will show results of its performance on the sample system used in this paper.

Applying the color model to the NPacs means that the original anchor colorimetries for which the ink-use optimization was computed, no longer represent the NPacs. Given the new, corrected colorimetries, a binning operation can be performed, whereby the available color gamut of a system is segmented (e.g. into sub-cubes or using k-means segmentation) and in each segment that has more than one candidate NPac, only the best NPac (that uses least ink) is kept. The result is a well distributed set of NPacs that each have the least amount of ink in their respective colorimetric neighborhood and from this set regular samplings can be re-interpolated with ease and compared against other color pipelines for performances.

To summarise this section, the optimization workflow is presented below:

```
00 Obtain NP colorimetries
01 Compute color model
02 Define colorimetric anchor points
03 Formulate optimization constraints

04 For each anchor point
05     Solve optimization (ink-use)
06     Apply color model

07 Segment new colorimetries
08 Pick NPac in colorimetry neighborhood
09 Print and measure to validate
10 Evaluate ink-use on regular grid
```

Ink-use optimal NPacs

In this section the previously described optimization framework is applied to a representative printing system and detailed results will be shown. The sample system used has 6 inks (CMYKcm) and the ability to deposit 4 pixel states per each of the inks, resulting in $4^6 = 4096$ NPs.

A subset of these 4096 NPs was chosen first using the following heuristics: no more than 3 inks used at a time and no more than 6 drops of ink used at a time, resulting in 1424 NPs. Since ink-use optimization is the objective here, these constraints are safe to assume not to affect optimality. Samples using these 1424 NPs were printed and measured first, before a further reduction in the number of NPs was performed by selecting the smallest number of NPs that represent the same colorimetric volume and that don't impact ink-use optimality (i.e. if an NP can be replaced at the same colorimetry with others that use the same or less amount of ink, then it is assume to be safe to remove). The resulting number of NPs is 79 and they are plotted in Figure 4.

A color model, that predicts reflectance and colorimetry from NPacs, has been computed for this set-up with the following accuracy (tested on data independent of the training data):

Table 1: Color model accuracy

	ΔE 2000
median	1.38
95th %tile	3.81

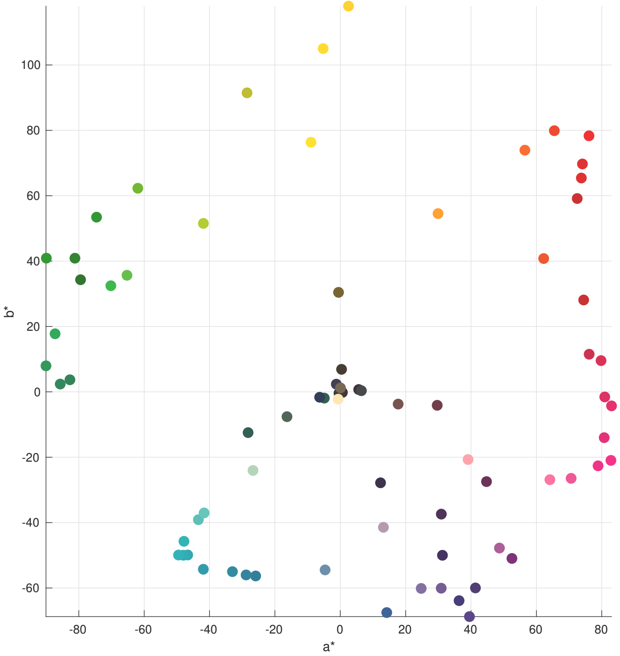

Figure 4. *97 NP colorimetries for the system of CMYKcm inks up to 3 drops per pixel (as described in the text), shown as an a*b* projection in CIE Lab space.*

The anchor points are then generated as those CIE Labs that are inside the available color gamut defined by the NP colorimetries, selected from a regular grid sampling – in the exmples below a 199^3 sampling was used resulting in 7,880,599 gamut interior Labs. Note that the higher this sampling the closer it represents the actual gamut of the device. Figure 5 then shows a subset of 17,402 of these anchor points chosen for the optimization – these are Lab-regular samples spread within the available gamut of the system. Here a consideration to take into account is whether to include samples that lie on the hull of the system or whether to only use samples that are strictly inside. With a high density sampling as used here (e.g. 199^3) the difference will be negligible since the interior samples contain ones that are very close to the gamut hull, within print and measure repeatability of the system.

The optimization itself is then performed in a Yule-Nielsen corrected XYZ space, hence both the NP colorimetries from Fig. 4 and the anchor points from Fig. 5 are represented in this domain. The optimization – the linear programming approach outlined earlier (using MATLAB's optimization toolbox or python scipy's scipy.optimize.linprog) is then executed for all the 7.9M anchor points above and the result is a list of ink-optimized NPacs.

For a subset of the samples used, Fig. 6 plots the colorimetries predicted using the color model compared against the colorimetries obtained from printing and measuring them. While they clearly depart from the previous synthetic anchor colorimetries the differences are relatively small and in line with the expected color model accuracy performance. The color difference of predicted vs measured, evaluated using the CIE ΔE 2000 metric are also reported in Table 2.

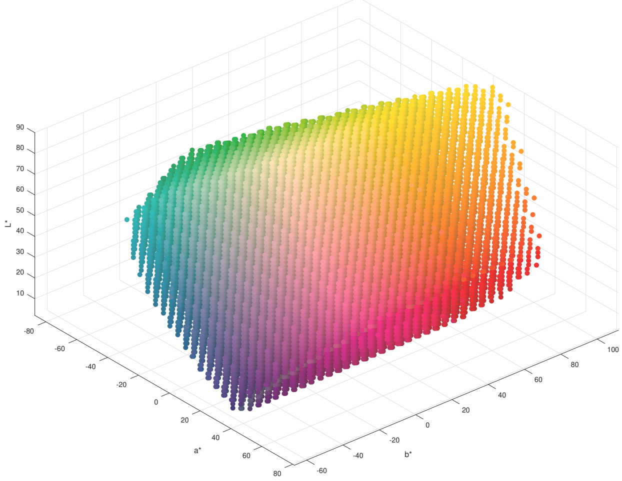

Figure 5. *Anchor points for ink-use optimization plotted in CIE Lab. The plot above shows a sub-set of the full set of anchor points.*

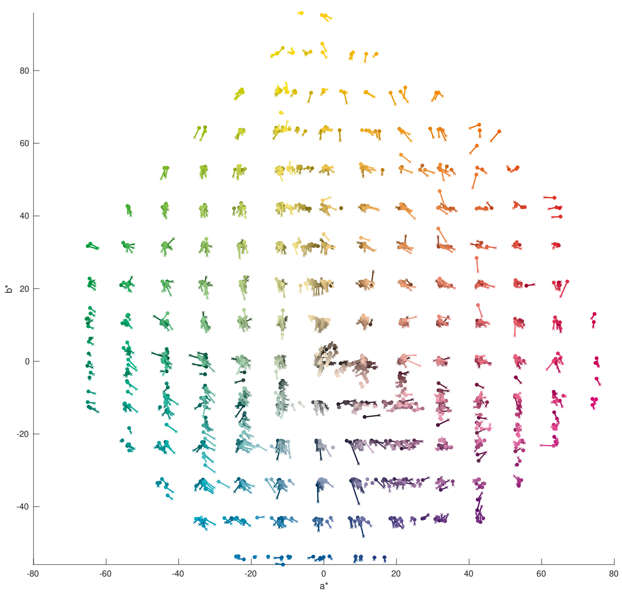

Figure 6. *Color model corrected colorimetry vs measured colorimetry distribution of minimum ink-use optimized NPacs.*

Table 2: Optimized NPacs predicted color accuracy

	ΔE 2000
median	1.04
95th %tile	2.21
max	3.54

Fig. 6 shows a reduced set of 1588 Labs out of the original 7,9M samples since the aforementioned segmentation has been performed at this stage already, using a regular cube binning of density 25^3, picking the best ink-use NPac in each case and thereby reducing the number of samples used next. Figure 7 below instead shows the ink-use distribution, where markers indicate the relative amount of ink used by the NPac corresponding to the respective colorimetry – a smaller marker indicates less ink used. For comparison, the maximum ink-use case is also shown in the same figure (bottom). This can be had simply by minimizing the negative of the ink-use function shown in Eq. 14. Note that the results plotted below do not correspond to the global most amount of ink that could be used to match a given colorimetry since not all NPs were available to the optimization and since the earlier pre-filtering of NPs (e.g. from 4096 to 1424 to 79) already considered minimum ink-use as the target metric and discarded NPs that could use more ink.

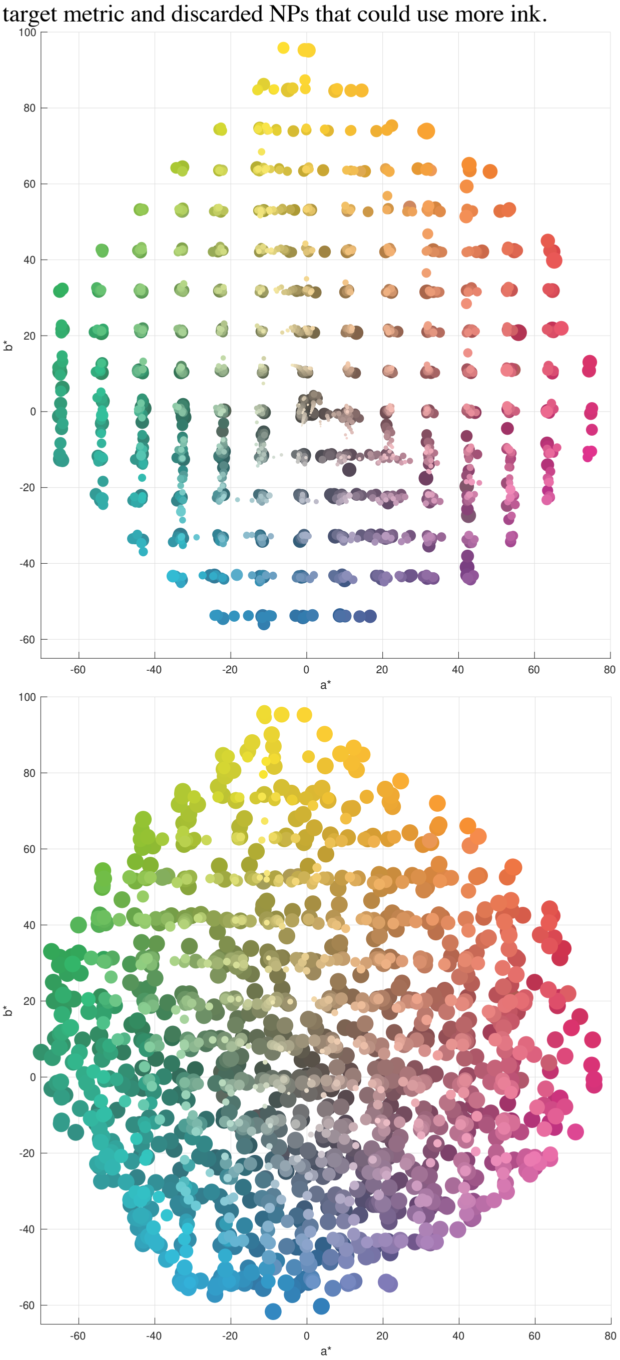

Figure 7. *Optimization result after binning (marker size proportional to amount of ink used by NPac at a given colorimetric location) – showing least amount of ink (top), and most amount (within the constraints of the # of NPs used here).*

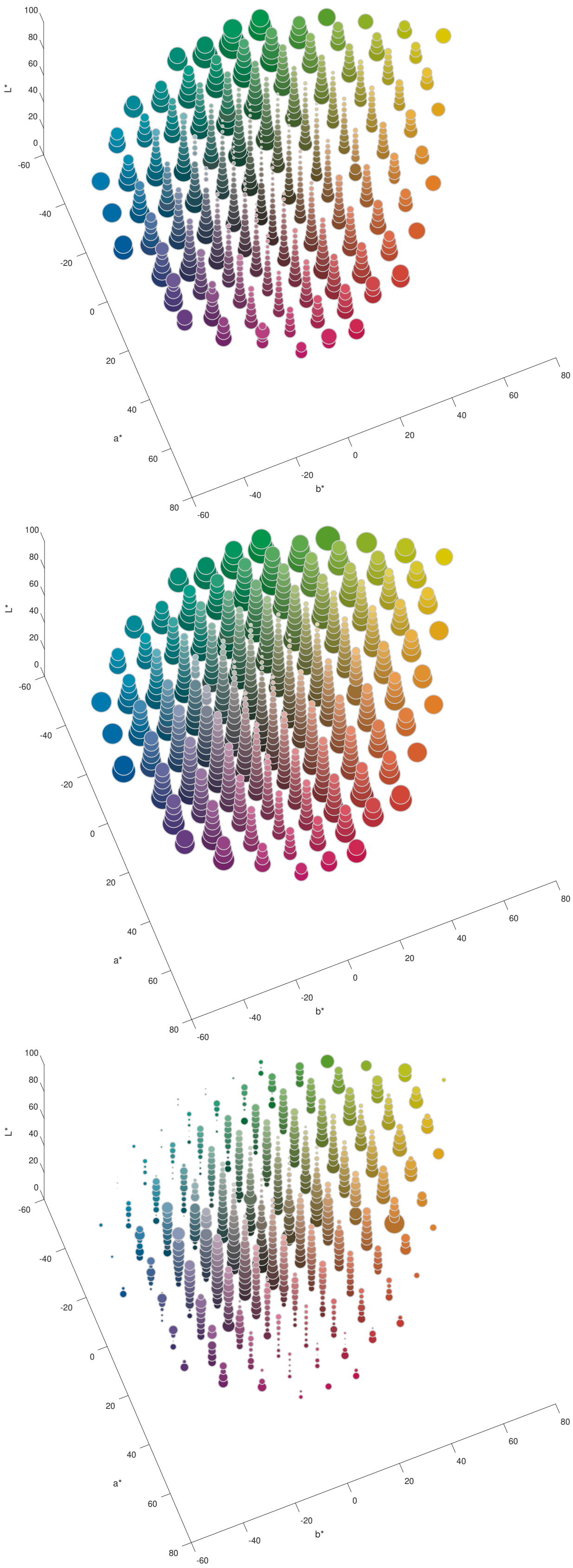

Figure 8. *Optimization result over re-interpolated uniform LAB grid (top) compared against a traditional ink-channel based pipeline with maxGCR profiling (centre) and the relative difference between the two – relating to the results shown in Table 3.*

Table 3: Ink-use and computation time results for current vs previous optimization approaches vs traditional pipeline

	Domain: # of NPs	Evaluated: # of NPacs	Ink-use (total)	Ink-saving (vs maxGCR)	Computation time
maxGCR (traditional)	250	–	15314	–	–
Standard optimization	97	112,613,370	12707	-17.0%	6 hours
Random only	97	95,075,798	12807	-16.4%	4 hours
Direct (LP)	**97**	**7,880,599**	**12521**	**-18.2%**	**14 min**

Finally, a sparser, regular uniform LAB grid was generated to be used for ink-use evaluation as compared to another system using a traditional ink-vector based pipeline. In the case of the current optimized results, this LAB grid is simply re-interpolated (in Yule-Nielsen corrected XYZ space), and can be compared against any other system by interpolating the same LABs through it and obtaining per-patch ink-use data (e.g. by counting drops from a halftone image). The results over this regular LAB grid are shown in Figure 8, against an existing, traditional ink-vector based pipeline using maximum Gray Component Replacement – the most amount of black possible to use in current systems, which in turn results in the most ink-efficient pipeline (given the constraints of ink-vector based pipelines), as well as the relative ink-use improvement. Table 3 below shows the numeric results of both the data plotted in Figure 8 (maxGCR traditional, Direct LP) as well as two previous methods [3, 4].

Discussion & Conclusions

The optimization described earlier and applied to the simple case of ink-use has shown that a direct NPac generation is able to get to optimal results, also as compared to previous methods which it outperforms slightly, however at a significant performance gain (14 minutes vs 6 hours) and with added flexibility. There are two important aspects to this. The first is that the optimization can be executed over near-arbitrary dimensional domains (e.g. 1424 NPs or even the full 4096), for which a convex hull computation would not be possible with today's state of the art and for which the previous full or random-only sampling would either not be feasible (in the case of full) or likely take much loner to converge. Second, that this approach results in direct generation of optimal NPacs, rather than a search-based method which means that e.g. the case of optimizing over 97 NPs (therefore in a 97D space) takes minutes to compute on common hardware and all resulting NPacs are optimal. The results in the previous section used For this particular case it is possible to compare against a previous approach [3, 4] which is why this set-up was chosen, so that a comparison in both ink-use and computation time can be done, shown in Table 3. For higher numbers of NPs some aspects of the methods in [3, 4] would not be feasible to use at all and therefore could not be compared.

Another important aspect of the presented optimization is that due to the systematic nature of the computation, all computed NPacs share similar structure. In fact, all LP optimized NPacs use at-most 4 NPs and it can be shown that this is provably the case (with the synthetic color formation assumption). More details about this property will be expanded on in the final paper.

While the presented example case used a simple optimization objective function, it can readily be seen that extensions to more complex cases are possible. In particular, the same types of intuitions as were used in some prior work on image quality and grain optimization – also by means of a reward/penalty function for each for the NPs, weighed by the respective area coverages, can directly be applied as well (and still keeps the optimization linear). Quadratic extensions can be thought of too, for example the simple case of minimizing distance to some previously found optimal NPac (distance in ND being a quadratic term), etc. At the same time, it's also clear that this approach has limitations and will not apply to any case. The focus of this work for the future will be both in adding

References

[1] Neugebauer H. E. J. (1937) Die theoretischen Grundlagen des Mehrfarbenbuchdrucks, Zeitschrift für wissenschaftliche Photographie, Germany, 36/4:73–89.

[2] J A C Yule, W J Nielsen, "The penetration of light into paper and its effect on halftone reproduction." 1951 TAGA Proceedings, p 65-76.

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

[4] Morovič P., Morovič J. and García–Reyero J. M. (2011) HANS: Meet The CMY Metamers, *IS&T/SID Color and Imaging Conference '11*, 7–11 November 2011, San Jose, CA, pp. 229–233.

[5] Morovič, J., Morovič, P., Rius, M. and García–Reyero, J. M. 2013. 8 vertex HANS: An ultra-simple printer color architecture, In *21st IS&T Color and Imaging Conference*, Albuquerque, NM, 210-214.

[6] Morovič, P., Morovič, J., Gondek, J. and Ulichney, R. 2017. Direct Pattern Control Halftoning of Neugebauer Primaries, In *IEEE Transactions on Image Processing*, submitted for publication.

[7] Morovič, P., Morovič, J., Tastl, I., Gottwals, M., Dispoto, G. (2019) Co-optimization of color and mechanical properties by volumetric voxel control, *Structural and Multidisciplinary Optimization*, pp. 1-14. https://doi.org/10.1007/s00158-019-02240-8

[8] Morovič P., Morovič J., Fariña X., Gasparin P., Encrenaz M., Arnabat J. (2015) Spectral and color prediction for arbitrary halftone patterns: a drop-by-drop, WYSIWYG, "ink on display" print preview, *23rd IS&T Color and Imaging Conference*, Darmstadt, Germany.

[9] Boyd. S., Vandenberghe, L., *Convex Optimization*, Cambridge University Press, ISBN 978-0-521-83378-3, 2004

[10] Chvatal, V., *Linear Programming*. W.H. Freeman. ISBN 978-0-7167-1587-0, 1983

[11] Fischler, M. A., Bolles, R. C., Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Comm. of the ACM*, Vol 24, pp 381-395, (1981)

[12] Chen, Y., Berns, R. S., Taplin, L. A., Extending Printing Color Gamut by Optimizing the Spectral Reflectance of Inks, *12th IS&T/SID Color Imaging Conference*, November 9, 2004, Scottsdale, Arizona, USA, 2004.

[13] Morovic J., Morovic P., (2017) HANS print smoothness optimization and continuous control, *25th IS&T Color and Imaging Conference*, Lillehammer, Norway

[14] Walsh J. L. (1921) On the Transformation of Convex Point Sets, *Annals of Mathematics*, Vol. 22, No. 4 (Jun., 1921), pp. 262-266.

Author Biography

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 150+ US patents filed (90+ granted) to

date. His interests include 2D/3D image processing, color vision, computational photography, computational geometry. His Erdős number is 4.

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.

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 150+ US patents (90+ granted).

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.

Tanausú Ramírez received his Ph.D. in Computer Architecture from the Universitat Politecnica de Catalunya (Spain) in 2009 and holds a BSc. and MSc. in Computer Science from the Universidad de Las Palmas de Gran Canaria (Spain). He has been working in HP since 2014 as a senior R&D SW engineer at the Large Format printing division, focusing on the investigation, development and deployment of color calibration algorithms and printing pipeline tools. Prior to joining HP he was a researcher at the Universitat Politecnica de Catalunya and research scientist at Intel Labs Barcelona. His research interests include color and image processing, computer vision and HW/SW architectures.

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.

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.