Printer model inversion by constrained optimization

Tomasz J. Cholewo

Software Research, Lexmark International, Inc.
740 New Circle Rd NW, Lexington, KY 40511
e-mail: cholewo@lexmark.com

ABSTRACT

This paper describes a novel method for finding colorant amounts for which a printer will produce a requested color appearance based on constrained optimization. An error function defines the gamut mapping method and black replacement method. The constraints limit the feasible solution region to the device gamut and prevent exceeding the maximum total area coverage. Colorant values corresponding to in-gamut colors are found with precision limited only by the accuracy of the device model. Out-of-gamut colors are mapped to colors within the boundary of the device gamut. This general approach, used in conjunction with different types of color difference equations, can perform a wide range of out-of-gamut mappings such as chroma clipping or for finding colors on gamut boundary having specified properties. We present an application of this method to the creation of PostScript color rendering dictionaries and ICC profiles.

Keywords: color conversion, gamut mapping, printer modeling, inverse modeling, constrained optimization

1. INTRODUCTION

A typical problem for color printers is how to reproduce the appearance of colors given a set of colorants. A specific combination of the amounts of each of the colorants constitutes a specification of color in a device-dependent space such as CMYK or RGB. However, this way of describing color is inherently non-portable between different printer models and different media. Therefore in recent years color management systems based on device-independent color spaces such as CIEXYZ and CIELAB have been gaining widespread use. Most of intelligence of such systems lies in device profiles which store the information about how to convert from a device-dependent to a device-independent color space (forward mapping) and how to perform the conversion in the opposite direction (inverse mapping). These mappings are usually stored in the form of multi-dimensional lookup tables, though they can also be represented in the form of a vector of parameters for a device model.

This paper describes a novel method for the creation of inverse mappings based on the concept of constrained optimization (also known as nonlinear programming). The error function implies the gamut mapping method and black replacement method. Constraints limit the feasible solution region to the device gamut and prevent exceeding the maximum total area coverage. We present an application of this method to the creation of Color Rendering Dictionaries (CRDs) by populating an interpolation table having CIELAB inputs and printer native colorant space (such as CMYK) outputs with inverted values. The same table is employed as part of a printer ICC profile.

Other researchers have previously used optimization methods to solve the inverse problem. Liang[1] developed a system for matching colors using lookup table device models. The model inputs were optimized for a fixed value of black input to reduce the difference between the predicted and desired colors. When the match was not found, the black level was adjusted, and the optimization process was reset and run again. The out-of-gamut colors were handled in the pre-processing stage, that mapped all the input colors to colors reproducible by the device.

Tominaga[2] used an inverse mapping method for direct training of a neural network controller for CIELAB to CMYK conversion. His method, however, does not allow for explicit black generation specification, and the behavior of colorants for out-of-gamut colors cannot be controlled. The tests were performed only for in-gamut colors and used the regular ΔE_{76} color differences.

Nakauchi et al.[3] performed image gamut mapping by cascading previously trained inverse (CIELAB to CMYK) and forward (CMYK to CIELAB) neural network models of the printer. Out-of-gamut color coordinates were simply clipped to the $[0, 1]$ range in the CMYK space which results in a gamut mapping not based on color perception. Backpropagation of the error derivatives through both models was used to find the input CIELAB values minimizing the output ΔE error.

2. DESCRIPTION OF THE METHOD

The first step in our method is the creation of a continuous model $\mathbf{f}$ of the printer which, for each value CMYK in the colorant space CMYK, calculates a corresponding value Lab in the CIELAB color space:

$$\text{Lab} = \mathbf{f}(\text{CMYK}, \boldsymbol{w})$$

Vector $\boldsymbol{w}$ represents model parameters. A number of different techniques can be used for implementing the function $\mathbf{f}$ such as Neugebauer equations, lookup tables with interpolation, neural networks, fuzzy logic, or polynomial regression.[4] The printer model is developed by printing a set of color patches with known colorant values CMYK_i (such as an IT8.7/3 chart) and measuring the corresponding color values of the patches Lab_{di}. The best fit model parameters $\boldsymbol{w}_f^*$ are found by minimizing some error criterion such as a sum of squared color differences:

$$\boldsymbol{w}_f^* = \arg\min_{\boldsymbol{w}_f} \sum_{i=1}^{N} \left[\Delta E(\mathbf{f}(\text{CMYK}_i, \boldsymbol{w}_f), \text{Lab}_{di})\right]^2$$

where N is the number of patches. $\arg\min_x f(x)$ denotes a value of argument x for which function $f(x)$ attains a minimum value. A diagram of the model creation step is shown in Figure 1a.

It is well documented that the CIELAB space is not perceptually uniform. This can present problems during gamut mapping because there exist colors with identical values of h^* coordinate having different perceived hue. This effect is most pronounced for the blue and red colors. Performing gamut mapping in a uniform color space such as MLAB[5] prevents hue shifts. We achieve that by converting the CIELAB measurement data to MLAB and creating a printer model with MLAB outputs. This allows us to use the error difference formulas based on the assumption of the perceptual uniformity.

Our goal is the generation of the inverse mapping for the printer which for each perceptual color value produces suitable colorant amounts. One method of obtaining the solutions is creating an inverse model which can then be queried for specific color values. An inverse model $\mathbf{h}$ is a vector function

$$\text{CMYK} = \mathbf{h}(\text{Lab}, \boldsymbol{w}_b)$$

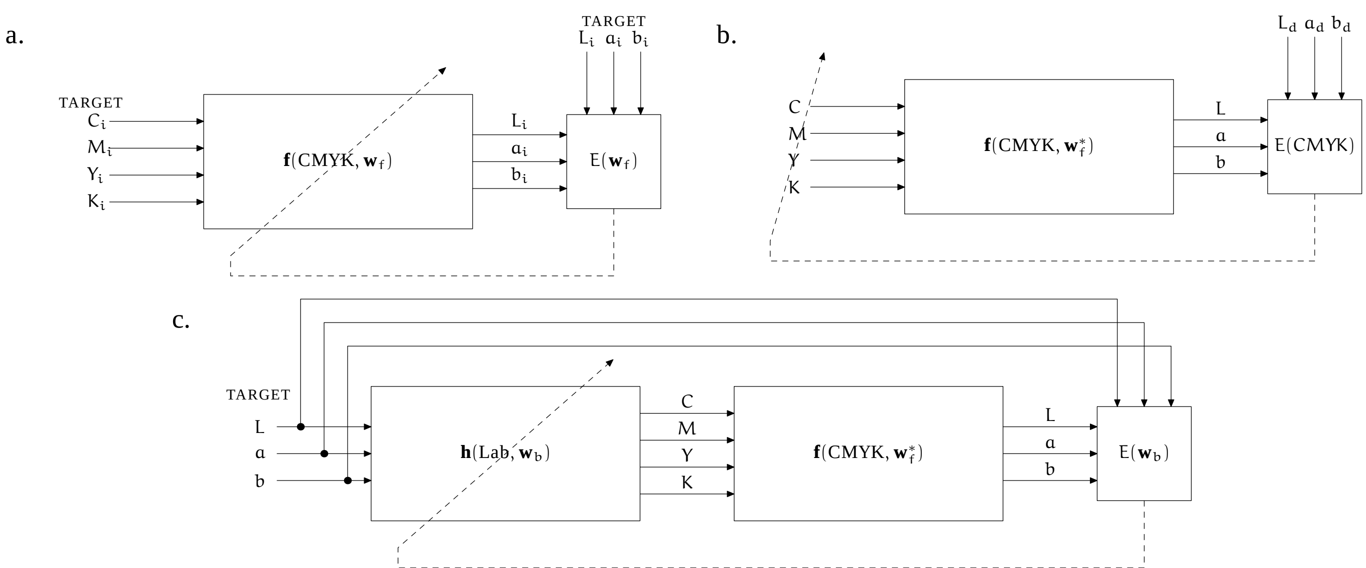

Figure 1. Block diagrams of a. printer model adaptation; b. inversion of a single color value; and c. creation of inverse printer model. The dashed arrow represents the process of adapting the model parameters ($\boldsymbol{w}_f$ or $\boldsymbol{w}_b$) or colorant values (CMYK) in order to minimize an error function $E(\cdot)$.

where $\boldsymbol{w}_b$ is a vector of parameters. These parameters are optimized by minimizing the error function E subject to constraints in the colorant space:

$$\boldsymbol{w}_b^* = \arg\min_{\boldsymbol{w}_b} \sum_i E(\mathbf{f}(\mathbf{h}(\mathrm{Lab}_i, \boldsymbol{w}_b), \boldsymbol{w}_f^*), \mathrm{Lab}_i).$$

Values Lab_i span the set of colors for which we want the parameters to be optimized. A block diagram of the optimization process is shown in Figure 1c. Next, the function $\mathbf{h}(\mathrm{Lab}, \boldsymbol{w}_b^*)$ is sampled to create a multi-dimensional lookup table.

In this paper we describe in more detail an alternative method of finding the inverse mapping by computing the colorant values for each color point individually. Its block diagram is shown in Figure 1b. In this case the desired colorant values CMYK_i^* are found for a particular color Lab_i by means of minimizing a cost function E related to the accuracy of color reproduction and gamut mapping with constraints reflecting the device gamut and printing process limitations:

$$\mathrm{CMYK}_i^* = \arg\min_{\mathrm{CMYK}} E(\mathbf{f}(\mathrm{CMYK}, \boldsymbol{w}_f^*), \mathrm{Lab}_i). \tag{1}$$

Figure 2 presents a simple comparison of unconstrained versus constrained optimization. The function E and the constraints are discussed in the following sections.

2.1. Color difference error function

The simplest form of color difference is the distance in the Cartesian CIELAB color space. Axes in this space are lightness (L^*) and two chromatic coordinates (a^* and b^*). We can apply different importance factors S to errors in each dimension to obtain a formula similar to a square of the well-known ΔE_{76} color difference equation:

$$E_{\mathrm{CIELAB}}(\mathrm{Lab}, \mathrm{Lab}_d) = \frac{1}{2}\Big[S_L(L^* - L_d^*)^2 + S_a(a^* - a_d^*)^2 + S_b(b^* - b_d^*)^2\Big].$$

The square root is omitted to make it easier to calculate the error gradient with respect to color coordinates:

$$\nabla_{\mathrm{Lab}} E_{\mathrm{CIELAB}}(\mathrm{Lab}, \mathrm{Lab}_d) = \Big[\frac{\partial E}{\partial L^*}, \quad \frac{\partial E}{\partial a^*}, \quad \frac{\partial E}{\partial b^*}\Big]^T = \Big[S_L(L^* - L_d^*), \quad S_a(a^* - a_d^*), \quad S_b(b^* - b_d^*)\Big]^T.$$

Gamut mapping is performed more conveniently in the perceptual coordinates of the CIELCH space. These cylindrical coordinates are lightness (L^*), chroma (C^*), and hue (h^*) which are obtained from the CIELAB coordinates using well-known formulas:

$$L^* = L^*; \qquad C^* = \sqrt{a^{*2} + b^{*2}}; \qquad h^* = \arctan\frac{b^*}{a^*}.$$

The error formula we use in this color space is similar to the ΔE_{94} CIE color difference equation:

$$E_{\mathrm{CIELCH}}(\mathrm{LCh}, \mathrm{LCh}_d) = \frac{1}{2}\Big[S_L(L^* - L_d^*)^2 + S_C(C^* - C_d^*)^2 + S_h(h^* - h_d^*)^2\Big].$$

Here the values of the S coefficients are not determined by the value of chroma coordinate but are chosen to obtain the desired weighing of each coordinate. These weights can vary depending on the desired color LCh_d.

The error gradient of the above error with respect to CIELCH coordinates is:

$$\nabla_{\mathrm{LCh}} E_{\mathrm{CIELCH}}(\mathrm{LCh}, \mathrm{LCh}_d) = \Big[S_L(L^* - L_d^*), \quad S_C(C^* - C_d^*), \quad S_h(h^* - h_d^*)\Big]^T.$$

For computational convenience and to avoid problems associated with the hue angle discontinuity, both the error and the gradient are expressed as functions of L^*, a^*, and b^*:

$$E_{\mathrm{CIELCH}}(\mathrm{Lab}, \mathrm{Lab}_d) = \frac{1}{2}\Big[S_L(L^* - L_d^*)^2 + S_C(C^* - C_d^*)^2 + S_h(C^*C_d^* - a^*a_d^* - b^*b_d^*)\Big].$$

The gradient then becomes:

$$\nabla_{Lab} E_{CIELCH}(Lab, Lab_d) = \left[S_L(L^* - L^*_d), \quad \gamma a^* - S_h a^*_d, \quad \gamma b^* - S_h b^*_d\right]^T$$

where

$$\gamma = S_C + \frac{C^*_d}{C^*}(S_h - S_C).$$

Similar equations were used by Tominaga[6] for printer model adaptation, but his formulation was based on trigonometric functions which are more expensive to calculate.

By setting $S_L = S_h = 1$ and $S_C = 0.1$ we can obtain a three-dimensional out-of-gamut mapping that resembles chroma clipping. Making $S_C > S_h \geq S_L$ results in a mapping that attempts to preserve the chroma at the cost of accuracy in hue and lightness, which can be used for creation of lookup tables for the ICC saturation rendering intent. Experiments performed by Katoh and Ito[7] on gamut mapping of computer generated images show that most observers prefer mappings performed with $S_L \geq S_h \geq S_C$. The weights S are related to the coefficients K used by Katoh: $S_L = 2/K_L^2$, $S_C = 2/K_C^2$, and $S_h = 2/K_H^2$.

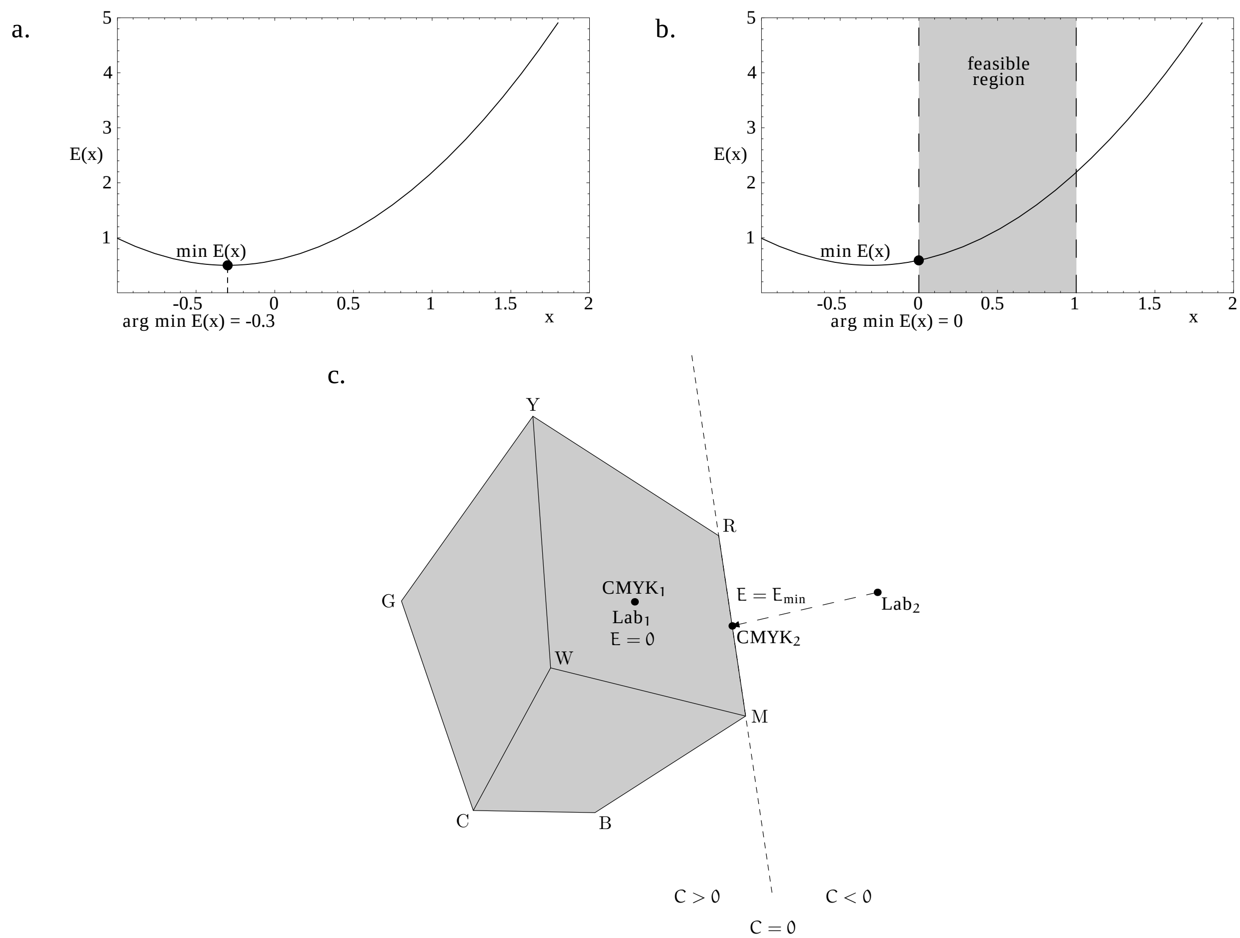

Figure 2. Examples of function optimization: a. unconstrained optimization of a function $E(x)$; b. optimization with a constraint $0 \leq x \leq 1$; and c. optimization of an error function $E(\mathbf{f}(CMYK, \boldsymbol{w}^*_f), Lab_d)$ with a constraint $C \geq 0$. Colorant values $CMYK_1$ corresponding to an in-gamut CIELAB color Lab_1 are found such that the color is reproduced exactly ($E = 0$). For an out-of-gamut color Lab_2 a point $CMYK_2$ on the gamut boundary for which E attains a minimum is found.

The specific values of coefficients S do not significantly influence in-gamut color mapping because for them $E_{\text{CIELCH}} \approx 0$ is attainable. However, even in this case they can come back into play if unrealistic values of K_d (see Section 2.4) are requested.

2.2. Physical gamut constraint

This constraint reflects the fact that one cannot request less than 0 percent or more than 100 percent of each of the colorants:

$$0 \leq C, M, Y, K \leq 1.$$

In the presence of natural boundaries,[8] points on the surface of the gamut may correspond to non-extreme values of colorants, i.e., the gamut constraint may not be activated.

2.3. Total area coverage constraint

Most printing processes have technological limitations that reduce the maximum total area coverage (TAC), i.e., the total amount of colorants printed at any point. For inkjet printers excessive amounts of ink may cause soaking and wrinkling of the paper. Electrophotographic printers may exhibit poor fusing when a toner layer is too thick. The SWOP web offset printing standard explicitly sets the TAC limit at 300%. Therefore an additional constraint is imposed which limits the effective gamut of a printer to colors that can be produced without exceeding the TAC:

$$C + M + Y + K \leq \text{TAC}.$$

It is easy to extend this constraint to handle colorants with different physical properties by using a weighted sum in the above equation.

2.4. Black preference error function

Printers using more than three colorants have additional degrees of freedom to represent a specified color. For example it is possible to render all colors strictly inside the CMYK printer gamut using infinitely many colorant combinations. We can select the preferred level of black replacement by means of an additional error term, for example:

$$E_{\text{black}}(\text{CMYK}, K_d) = \frac{1}{2}(K - K_d)^2.$$

Its gradient is:

$$\nabla_{\text{CMYK}} E_{\text{black}}(\text{CMYK}, K_d) = [0, \quad 0, \quad 0, \quad K - K_d]^T.$$

Hung[9] described a method for colorimetric black replacement. For each color the maximum ($K_{\max}$) and minimum ($K_{\min}$) amounts of black with which it can be reproduced are determined. In our method $K_{\max}$ and $K_{\min}$ are found by inverting the printer model with $K_d = 1$ and $K_d = 0$, respectively, and with S_{black} set to a small value. Then the final inversion is performed with K_d set to a value between these two numbers. All these steps take into account TAC constraints.

Conversion from a calibrated CMYK space such as SWOP to a device CMYK space is important for applications such as offset press emulation. The calibrated CMYK spaces are usually defined by a table of standardized measurements relating a particular CMYK value to a CIELAB value. This gives us a specific value of black K_d which we use for preserving the original separation settings during creation of calibrated CMYK to device CMYK mappings. Emulation based on ICC profiles loses information about the black separation when the data is converted through the three-channel CIELAB space.

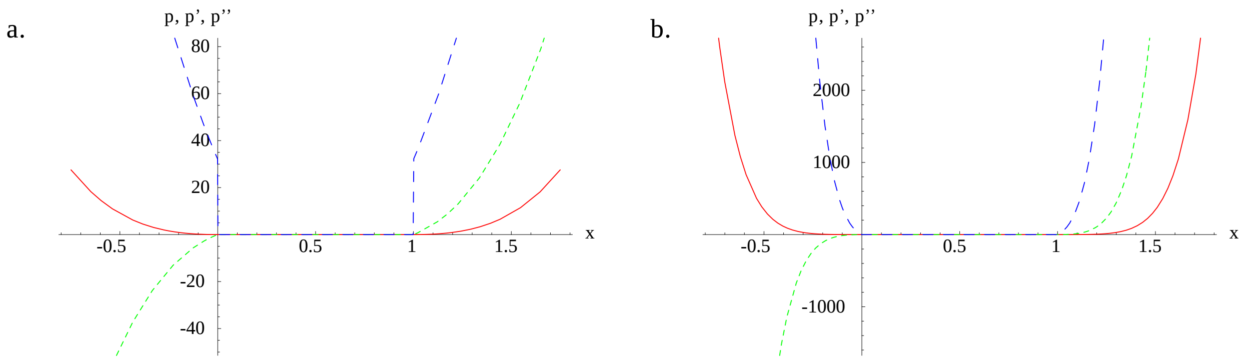

Figure 3. Penalty gap functions: a. biquadratic; b. bicubic. Continuous (red) line is the function $p(x)$, short-dashed (green) line shows its first derivative $p'(x)$, and long-dashed (blue) line represents the second derivative $p''(x)$.

3. SOLVING THE CONSTRAINED OPTIMIZATION PROBLEM

To minimize the error function in the presence of constraints, special optimization algorithms need to be used. For example, when only physical gamut constraints are specified we employed a bound-constrained optimization algorithm LBFGS.[10] However, for the inverse model method (shown in Figure 1c) the colorant space constraints become very complicated: $0 \leq \mathbf{h}(\mathrm{Lab}, \boldsymbol{w}_b)_i \leq 1$. Therefore we converted the constraints to additional error terms. This allows use of simpler unconstrained optimization algorithms. We use an efficient second-order optimization algorithm known as scaled conjugate gradient.[11] Each constraint is replaced by a penalty function which is continuous in the control space. This function should be equal to zero when the condition is satisfied (for example $0 \leq x \leq 1$) and greater than zero otherwise. A biquadratic penalty function (Figure 3a) is smooth and has a continuous first derivative:

$$p(x) = \begin{cases} 0 & \text{if } 0 \leq x \leq 1; \\ 16x^2(x-1)^2 & \text{otherwise.} \end{cases}$$

A bicubic function (Figure 3b):

$$p(x) = \begin{cases} 0 & \text{if } 0 \leq x \leq 1; \\ (|2x-1|^3 - 1)^3 & \text{otherwise.} \end{cases}$$

is needed when an optimization algorithm requires that the first derivative of the error function be smooth.

Using the penalty method the gamut constraint is converted to an error term:

$$E_{\text{gamut}} = p(C) + p(M) + p(Y) + p(K).$$

Its gradient is:

$$\nabla_{\text{CMYK}} E_{\text{gamut}} = \left[p'(C), \quad p'(M), \quad p'(Y), \quad p'(K)\right]^T.$$

For total area coverage constraint we obtain:

$$E_{\text{TAC}} = p\left[\frac{1}{\text{TAC}}(C + M + Y + K)\right]$$

and

$$\nabla_{\text{CMYK}} E_{\text{TAC}} = \left[\sigma, \quad \sigma, \quad \sigma, \quad \sigma\right]^T$$

where

$$\sigma = \frac{1}{\text{TAC}} p'\left(\frac{1}{\text{TAC}}(C + M + Y + K)\right).$$

The total error function $E(\mathrm{CMYK})$ consists of the error terms both in the color space (color difference error E_{CIELCH}) and in the colorant space (such as gamut and TAC constraint penalties and black preference error) and is parameterized by the desired color value Lab_d and the desired black level K_d:

$$E(\mathrm{CMYK}, \mathrm{Lab}_d, K_d) = E_{\mathrm{CIELCH}}(\mathrm{Lab}) + E_{\mathrm{CMYK}}(\mathrm{CMYK}) = E_{\mathrm{CIELCH}}(\mathbf{f}(\mathrm{CMYK}, \boldsymbol{w}_f^*), \mathrm{Lab}_d) + E_{\mathrm{CMYK}}(\mathrm{CMYK}, K_d).$$

The control space error function E_{CMYK} is constructed as a weighted sum of two constraint penalties and black preference error:

$$E_{\mathrm{CMYK}}(\mathrm{CMYK}, K_d) = S_{\mathrm{gamut}} E_{\mathrm{gamut}} + S_{\mathrm{TAC}} E_{\mathrm{TAC}} + S_{\mathrm{black}} E_{\mathrm{black}}.$$

The S coefficients determine the importance of error components, relative to each other and also relative to E_{CIELCH}. The gradient of the error function E_{CMYK} with respect to CMYK control variables is

$$\nabla_{\mathrm{CMYK}} E_{\mathrm{CMYK}} = S_{\mathrm{gamut}} \nabla_{\mathrm{CMYK}} E_{\mathrm{gamut}} + S_{\mathrm{TAC}} \nabla_{\mathrm{CMYK}} E_{\mathrm{TAC}} + S_{\mathrm{black}} \nabla_{\mathrm{CMYK}} E_{\mathrm{black}}.$$

Knowledge of the gradient vector of the total error function E with respect to colorant variables is needed for efficient solving of Equation (1). It can be found as:

$$\nabla_{\mathrm{CMYK}} E(\mathrm{CMYK}, \mathrm{Lab}_d, K_d) = \nabla_{\mathrm{CMYK}} E_{\mathrm{CIELCH}}(\mathbf{f}(\mathrm{CMYK}, \boldsymbol{w}_f^*), \mathrm{Lab}_d) + \nabla_{\mathrm{CMYK}} E_{\mathrm{CMYK}}(\mathrm{CMYK}, K_d).$$

The first term is then expanded by the derivative chain rule:

$$\begin{aligned}\nabla_{\mathrm{CMYK}} E_{\mathrm{CIELCH}}(\mathbf{f}(\mathrm{CMYK}, \boldsymbol{w}_f^*), \mathrm{Lab}_d) &= \nabla_{\mathrm{CMYK}}(\mathbf{f}(\mathrm{CMYK}, \boldsymbol{w}_f^*), \mathrm{Lab}_d) \cdot \nabla_{\mathrm{Lab}} E_{\mathrm{CIELCH}}(\mathrm{Lab}, \mathrm{Lab}_d) \\ &= \mathbf{J}_\mathbf{f}^T \cdot \nabla_{\mathrm{Lab}} E_{\mathrm{CIELCH}}(\mathrm{Lab}, \mathrm{Lab}_d)\end{aligned}$$

where $\mathbf{J}_\mathbf{f}$ is the Jacobian matrix of the forward model $\mathbf{f}$, $\mathrm{Lab} = \mathbf{f}(\mathrm{CMYK}, \boldsymbol{w}_f^*)$, and $\nabla_{\mathrm{CIELAB}} E_{\mathrm{CIELCH}}(\mathrm{Lab}, \mathrm{Lab}_d)$ was derived in Section 2.1. The Jacobian has to be determined, usually numerically, for the specific type of the function $\mathbf{f}$ used. The previous equation can be written in an expanded form:

$$\begin{bmatrix} \frac{\partial E}{\partial C} \\ \frac{\partial E}{\partial M} \\ \frac{\partial E}{\partial Y} \\ \frac{\partial E}{\partial K} \end{bmatrix} = \begin{bmatrix} \frac{\partial L^*}{\partial C} & \frac{\partial a^*}{\partial C} & \frac{\partial b^*}{\partial C} \\ \frac{\partial L^*}{\partial M} & \frac{\partial a^*}{\partial M} & \frac{\partial b^*}{\partial M} \\ \frac{\partial L^*}{\partial Y} & \frac{\partial a^*}{\partial Y} & \frac{\partial b^*}{\partial Y} \\ \frac{\partial L^*}{\partial K} & \frac{\partial a^*}{\partial K} & \frac{\partial b^*}{\partial K} \end{bmatrix} \begin{bmatrix} \frac{\partial E}{\partial L^*} \\ \frac{\partial E}{\partial a^*} \\ \frac{\partial E}{\partial b^*} \end{bmatrix}.$$

Each of the partial derivatives is evaluated at a specific input value CMYK.

The process of optimization stops when it reaches a stationary point on the error surface. At such a point, by definition, $\nabla E = 0$, that is all the "gradient forces" contributed by different error terms are in balance. Since the gamut and limit penalties can be made arbitrarily large, they can effectively trap the solution inside the gamut of the printer. This way the type of out-of-gamut mapping is determined mainly by the E_{CIELCH} term.

As in all global optimization problems, there exists a possibility of getting trapped in a local minimum of the error function. However, experiments show that this is rarely the case for color mappings arising in printer modeling because the relationships between the colorant amounts and the resulting color are predominantly monotonic and smooth. Suboptimal solutions can be detected by comparing the result with the values obtained for the neighboring colors and corrected by restarting the optimization process with different initial conditions.

4. CONCLUSIONS

We presented a method for creating inverse color mappings. An example of such a mapping created for a Lexmark Optra Color 45 inkjet printer and a sparse grid of CIELAB values is shown in Figure 4. The out-of-gamut mapping was performed in the MLAB space to avoid hue shifts caused by the non-uniformity of CIELAB.

A general formulation of our procedure makes it easy to apply it to a wide range of color mapping problems. The method can be extended to handle more colorants, such as for Hexachrome or CMYKOG processes, by adding additional constraints limiting the dimension of the solution space. It is possible to specify different out-of-gamut a.

b.

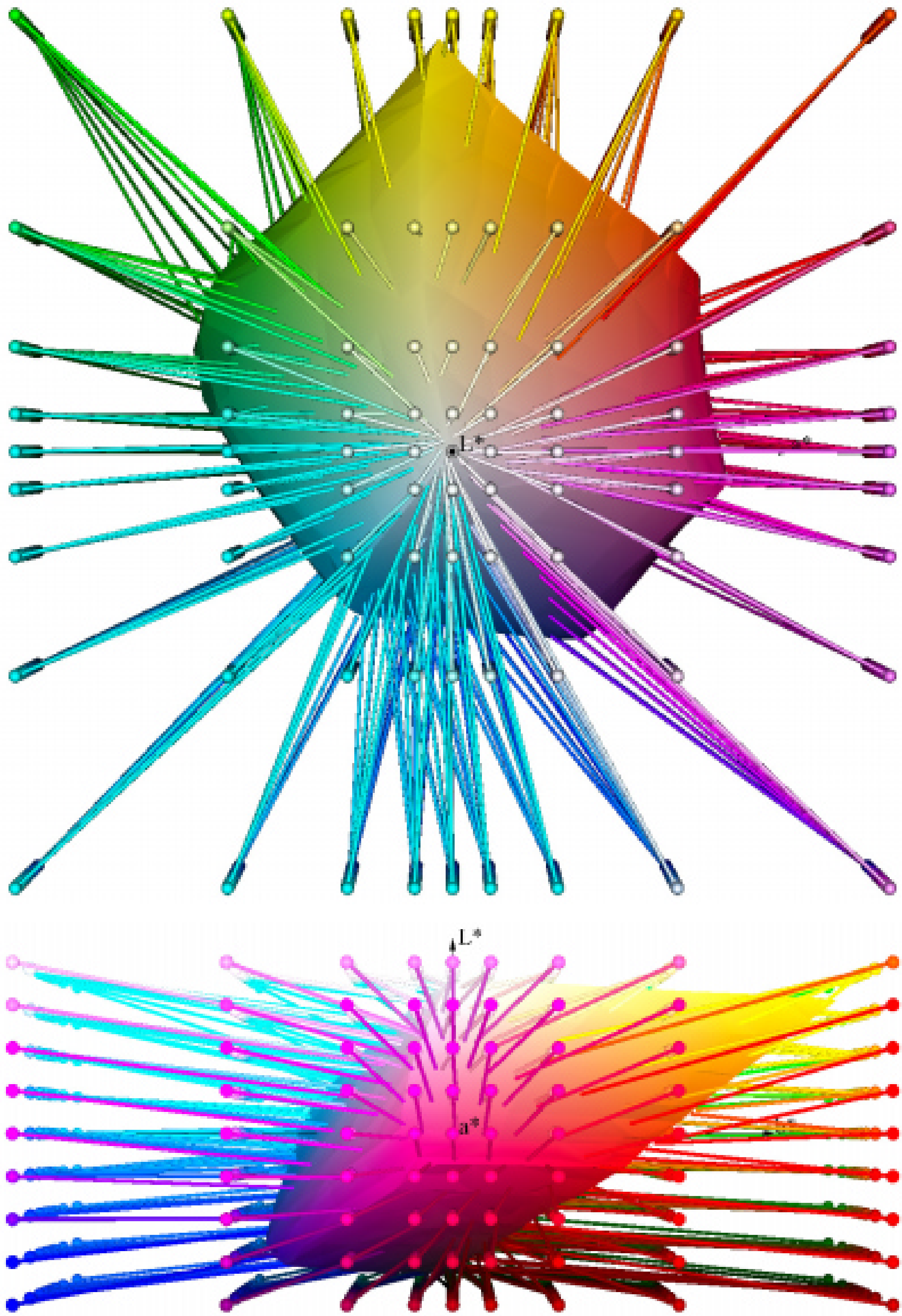

Figure 4. Visualization of inverse mapping between a $9 \times 9 \times 9$ non-uniform grid of CIELAB values and Lexmark Optra Color 45 inkjet CMYK values: a. a^*b^* view; b. L^*b^* view. Spheres marking the CIELAB grid points are connected with lines to the CIELAB values corresponding, according to the forward model $f(\cdot)$, to CMYK values obtained from the mapping.

mapping algorithms just by changing the error function terms. Our method does not depend on geometric gamut constraints and this allows for gamut shapes with process black color darker than real black and for natural boundaries. Basing all the calculations on a continuous printer model permits the inversion algorithm to operate in the conceptually continuous colorant space. This avoids discontinuities introduced by geometrical inversion methods based on searching for color space points in the neighborhood of the desired color. The optimization process can be easily applied for finding colors on the gamut boundary with defined properties, for example having specified chroma and hue and minimum lightness.[12] Tables that indicate whether a given color is inside a gamut of a device can be constructed by noticing that the total error for all in-gamut colors will be very close to zero. Smooth gloss of printouts can be assured by introducing a penalty term for solutions with small total toner amounts.

REFERENCES

1. Z. Liang, "Generic image matching system (GIMS)," in *Proceedings of SPIE*, vol. 1670, pp. 255–265, (San Jose, Calif.), 1992.
2. S. Tominaga, "A color mapping method for CMYK printers and its evaluation," in *Proceedings of the IS&T/SID Color Imaging Conference*, pp. 172–176, (Scottsdale, Ariz.), November 1996.
3. S. Nakauchi, M. Imamura, and S. Usui, "Color gamut mapping by optimizing perceptual image quality," in *Proceedings of the IS&T/SID Color Imaging Conference*, pp. 63–67, (Scottsdale, Ariz.), November 1996.
4. H. R. Kang, *Color technology for electronic imaging devices*, SPIE Optical Engineering Press, Bellingham, Wash., 1996.
5. P.-C. Hung, "Color spaces for color-gamut mapping," *Journal of Electronic Imaging* **8**, pp. 354–364, October 1999.
6. S. Tominaga, "Color conversion using neural networks," in *Proceedings of SPIE*, pp. 66–75, (San Jose, Calif.), January 1998.
7. N. Katoh and M. Ito, "Gamut mapping for computer generated images," in *Proceedings of the IS&T/SID Color Imaging Conference*, pp. 126–129, (Scottsdale, Ariz.), November 1996.
8. M. Mahy, "Insight into the solutions of the Neugebauer equations," in *Proceedings of SPIE*, vol. 3300, pp. 76–85, (San Jose, Calif.), January 1998.
9. P.-C. Hung, "Smooth colorimetric calibration technique utilizing the entire gamut of CMYK printers," *Journal of Electronic Imaging* **3**, pp. 415–424, October 1994.
10. C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "LBFGS-B: Fortran subroutines for large-scale bound constrained optimization," Tech. Rep. NAM-11, EECS Department, Northwestern University, 1994.
11. M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks* **6**, pp. 525–533, 1993.
12. T. J. Cholewo, "Black replacement using lightness scaling," in *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Science and Technology*, (San Jose, Calif.), January 2000.