A Color Mapping Method for CMYK Printers and Its Evaluation

Shoji Tominaga
Osaka Electro-Communication University
Neyagawa, Osaka, Japan

Abstract

A method is described for realizing the mapping from the three-dimensional color space of color stimuli to the four-dimensional CMYK space of printer ink signals. Color reproduction on a printer is considered as the problem of controlling an unknown static system with four inputs and three outputs. The algorithms for the mapping are based on a two-phase procedure using neural networks, which eliminate the need of UCR and GCR. The first phase determines the printer model with a neural network, and the second phase determines the controller to realize the inverse mapping for the printer model. The performance of the proposed method was examined on experiments using a dye sublimation printer and an ink jet printer.

Introduction

Color reproduction on a printer requires color conversion between the color signals, depending on the printer, and the standard color coordinates, representing color appearance.[1-2] This color conversion means the problem of estimating the amounts of primary inks necessary to produce a desired color stimulus on a printer. There are several methods for solving this problem. These are (1) an analytical method using the Neugebauer equations, (2) matrix transformation, (3) use of a look-up table and three-dimensional interpolation,[3] and (4) a neural network method.[4-5] In (2)-(4), a color printer is regarded as a black box.

A neural network is convenient for modeling a nonlinear transformation between two color spaces, for which a mathematical description is difficult. However most neural network methods were developed to control only three color signals CMY of printer primaries. We need the four inks of CMYK for high quality color reproduction. In this case, the conversion from the color stimulus values to the printer color signals CMYK is not unique because it has to determine the mapping from a three-dimensional color space to a higher four-dimensional space.

In the previous talk,[6] the author proposed a solution method for the mapping from the three-dimensional color space of color stimuli to the CMYK color space of printer signals. The CIE-L*a*b* color system is used as the color space. In this method we regard a color printer as an unknown static system with four inputs and three outputs. We determine the CMYK values as the input control signals so that the printer system outputs the desired L*a*b* values. This control signal is determined by a neural network method based on a two-phase procedure.

Because our method provides a direct mapping from the L*a*b* color space to the printer CMYK space, we can eliminate the need to do undercolor removal (UCR) and gray component replacement (GCR). The basic principle for solving the control problem for the direct mapping was presented in the previous talk. This paper presents practical algorithms for realizing the mapping by using neural networks. We evaluates the performance on experiments using a dye sublimation printer and an ink jet printer.

Mapping Method

Principle

Color reproduction must determine the digital values of the ink signals CMYK to generate any desired L*a*b* color specification within the gamut of a printer, that is, to realize the mapping from the CIE-L*a*b* color space to the printer CMYK color space. We consider the color reproduction problem as the problem of controlling a system with unknown characteristics. Figure 1 shows a total network structure combining both networks for modeling a color printer and for controlling the printer.

The color printer is an unknown static system with four inputs and three outputs. A controller tries to realize the inverse mapping of the system. Therefore, the controller must determine the printer ink signals so as to minimize the color stimulus error in the L*a*b* color specification system. Even if we try to make the direct mapping from the L*a*b* color space to the printer CMYK color space by using a neural network, the iterative learning algorithms for determining the network parameters do not converge because the mapping is defined from the three-dimensional space to a higher dimensional space. We propose a method for finding this mapping solution based on a two-phase procedure.

Determination of the Printer Model

In the first phase, to identify the unknown system of a printer, the system is modeled using a neural network. To do this, we first measure many color patches which are output from the printer in grid points sampled uniformly in the whole CMYK space. The data set of these measured L*a*b* values from printed colors and the corresponding CMYK values from the printer inputs is used as the training data of the network.

The upper part of the total network in Figure 1 models the printer. The mapping from the four ink signals CMYK to the L*a*b* space is described using a four-layered network. Each of two hidden layers has 10 units. In Figure 1 the symbols of filled circles and filled squares indicate, respectively, the units and biases to be determined in the first phase. The 4-10-10-3 type network for modeling the printer has 170 weights and 23 biases.

All the input/output signals of the printer model are normalized. First, the CMYK values, which lie in the region of 0 to 255, are scaled to the range [0, 1]. Concerning the output signals, we assume that the ranges for three quantities L*, a*, and b* are [0, 100], [-100, 100], and [-100, 100], respectively. The color gamut of the present printer is contained within the rectangular prism by these ranges. All of these ranges are then normalized into the range [0, 1].

Let o_i be the output of the unit i in the prior layer, w_{ji} be the weighting coefficient of connection from unit i to the target unit j, and b_i be the bias term of the unit. The input to unit j is then described as the sum of the weighted outputs from the prior layer.

$$\mathrm{net}_j = \sum_i w_{ji} o_i + b_j \tag{1}$$

The nonlinear output of unit j is

$$o_j = f(\mathrm{net}_j) \tag{2}$$

where f is the sigmoidal activation function

$$f(\mathrm{net}) = 1/\{1 + \exp(-4\alpha\ \mathrm{net})\} \tag{3}$$

This function takes any real number in the interval [0, 1], and the positive constant a represents the slope of f at net = 0. The operations of (1)-(2) are executed at all units except the input layer of the printer model.

The parameters of weights and biases are determined by a learning procedure of error backpropagation.[7] Now let us define the pth vector of the normalized L*a*b* values in the training data set as,

$$\mathbf{t}_p \equiv [t_{p1},\ t_{p2},\ t_{p3}]$$

and define the corresponding the output vector from the network as

$$\mathbf{o}_p \equiv [o_{p1},\ o_{p2},\ o_{p3}]$$

All the weights and biases in the network are then adjusted to minimize the squared error between the target $\mathbf{t}_p$ and the actual output $\mathbf{o}_p$, $E_p = \|\mathbf{t}_p - \mathbf{o}_p\|^2$. The change Δw_{ji} in the weight w_{ji} is computed by the following recursive algorithm including a momentum term:

$$\Delta w_{ji}(n+1) = \eta \delta_{pj} o_{pi} + \beta \Delta w_{ji}(n), \tag{4}$$

where n indicates the nth steps of learning. The notation $\Delta w_{ji}(n)$ represents the amount of correction in the weight from unit i to unit j in the next layer at the nth step. Moreover δ_{pj} is an error term in unit j, which is calculated recursively from the output layer by using the equations of error backpropagation (see Ref. 8). The notations η and β are positive constants called the learning constant and momentum constant, respectively. The change Δ_{bj} in the bias b_j is computed in the same manner as are the other weights.

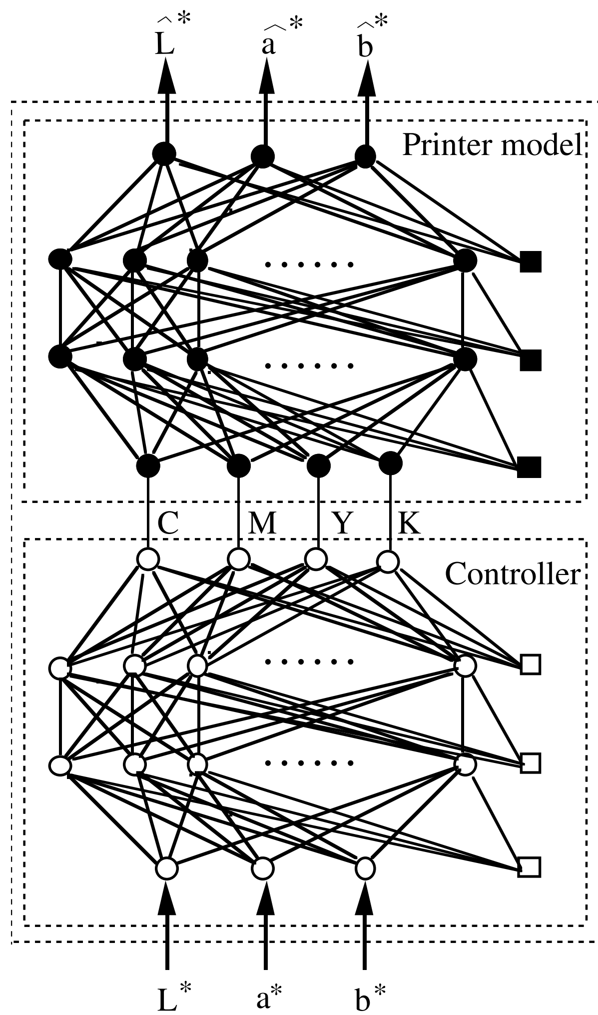

Figure 1. Total network structure for modeling a color printer and for controlling the printer.

Determination of the Controller

In the second phase, the control system of mapping a color specification vector of L*a*b* into an ink vector of CMYK is determined by a neural network technique. We adopt an indirect manner to perform this. Let us consider the total network system combining the controller and the printer as shown in Figure 1. From a global point of view, this total system represents a mapping from a target L*a*b* value to the output L*a*b* value of the printer. This mapping is an identity mapping to produce the same color as the input.

Because the network model for the printer was already obtained in the first phase, in the present phase we determine the identity mapping for the total network system. The structure and all parameters in the printer model are fixed, while network parameters in the controller are learned iteratively for the error minimization. When the total network system performs the one-to-one mapping, the network of the controller part must realize an inverse mapping of the printer model.

The controller is constructed with the four layered network of a 3-10-10-4 type as shown in the lower part of the Figure 1. The total network is an eight-layered network, consisting of the input/output layers and six hidden layers. Note that the 3rd and 4th hidden layers are double, where the same signals are copied simply from the lower layer to the upper. Therefore, only five hidden layers is effective in defining the whole mapping. Both the input/output signals of the total network are the normalized L*a*b* values in the range [0, 1]. The network learning is performed according to the following procedure.

(1) Because of the one-to-one mapping, the same L*a*b* values are presented to the input/output layers. No CMYK values are used as the training data.

(2) The total system is learned based on the rule of error-backpropagation. Here the weights and biases in the printer model part are fixed as the previously estimated values, which are never changed as $w_{ij}(n)=w_{ij}(0)$ and $b_j(n)=b_j(0)$. Then the parameters to be newly determined in the total system are limited to 194 parameters of 170 weights and 24 biases in all. In Figure 1, the units and biases shown with the symbols of open circles and squares are determined by learning the total network, starting from any random numbers.

(3) The error is propagated backward through the total network from the output layer to the input layer. The upper half does not change its network parameters, but only the error passes through the hidden layers. On the other hand, the lower half corrects its parameters by the recursive computations.

(4) When the total network system completes the mapping from the L*a*b* space to the L*a*b* space, the lower half of the total system must realize the inverse mapping of the printer model, that is the mapping from the L*a*b* space to the CMYK space.

Thus the desired controller of the printer is obtained by taking out the lower half of the total network.

Experiments

Non-impact printers of the thermal transfer-dye sublimation type and the ink jet type are used to examine the performance of the proposed method.

Printer Model

To make the printer model, we measured 6561 ($=9^4$) color patches printed by four inks CMYK under D_{65}. The color specifications of white papers used in both printers are used as the white standards X_0, Y_0, and Z_0. The training data set consist of 6561 pairs of the input ink signals and the corresponding L*a*b* color specifications of the printer outputs. All the data are normalized into the interval [0, 1].

The networks are simulated on a SUN SPARC Station 20. The initial values $w_{ij}(0)$ and $b_j(0)$ are set to random numbers in the interval [-0.5, 0.5]. The training data are presented randomly. This data presentation is not a simple random sampling among the data set, but it is based on a uniformly random sampling so that every element of the data set is selected once in a random order in one epoch. The learning rate and momentum constant are set to $\eta = \beta = 0.9$ at start, and in the iterative process, these constants are decreased at proper intervals. After 40000 iterations (epochs) of the iterative learning, the total system error for all the training data converges into a small value, and the learning process was terminated.

Figures 2 and 3 show the learning behaviors, that is, the convergence process of the system errors for dye sublimation type and ink jet type, respectively. Each curve with the symbol (a) represents the variation of the total squared error for all the training data as a function of the number of iteration. The errors at 40000 iterations are the small numbers of 1.05 for the dye sublimation type and 1.75 for the ink jet type.

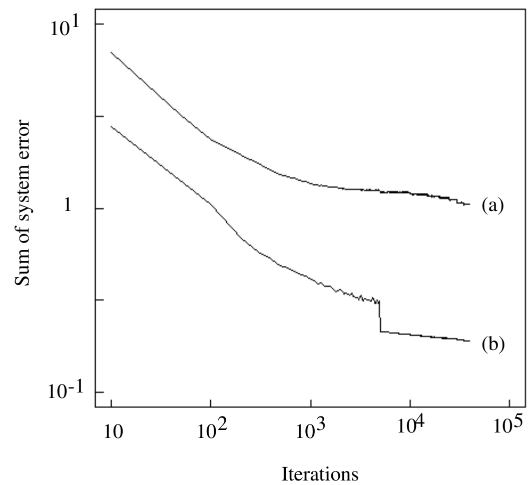

Figure 2. Learning behavior for dye sublimation. (a) printer model, (b) identity mapping.

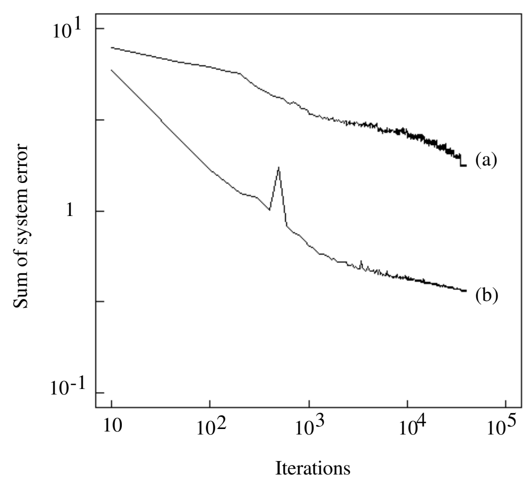

Figure 3. Learning behavior for ink jet. (a) printer model, (b) identity mapping.

Controller

To make the controller, the total eight-layered network, including the printer model obtained above, was trained according to the procedure in the previous section. We use the same L*a*b* data of 6561 training data as in making the printer model. For training the total network to realize the identity mapping, we present the same L*a*b* values to the input/output layers. Two curves with the symbol (b) in Figures 2 and 3 represent the learning behaviors in realizing the identity mapping for the dye sublimation type and ink jet type, respectively. Training the total network converges at around 40000 iterations. At this time the total squared errors for all the normalized data are 0.20 for the dye sublimation type and 0.37 for the ink jet type. Finally the controller part is extracted from the finished total network with eight layers.

Accuracy Test

The accuracy of color reproduction by the controller was examined. We use 148 color samples as the test data. To make these test data, first, we select 132 coordinate points uniformly in the input signal space of CMYK with K = 0, 128, and moreover select 16 points on the gray scale with C, M, Y = 0 in the CMYK space. Second, we print 148 color patches by using these selected CMYK values. The color patches are the target color samples for the accuracy test. These measurements present the L*a*b* color specifications of the target color stimuli.

To reproduce these target colors, first, we input the L*a*b* values to the controller, so that we get the CMYK ink values necessary to generate the target L*a*b* values. Second, we print color patches by applying these CMYK values to the real printers. The color specifications of the reproduced color samples are obtained by measuring the printed patches.

The accuracy is examined by comparing the target color specifications and the reproduced color specifications in the L*a*b* color space. The average color differences are 2.0 for the dye sublimation type, and 2.9 for the ink jet type. The maximum errors are 7.5 and 10.5, respectively.

Comparison with other methods

First, we have applied the interpolation method[3] to the color reproduction test, which uses a look-up table by interpolating the three-dimensional table data. A simulator of color conversion based on the prism interpolation is executed on a SUN SPARC Station. The set of the table data consists of 729 color samples which are produced at every 32 step of each scale of CMY. These three-dimensional color coordinates are interpolated using a triangular prism. The reliability of this look-up table is examined using the same test data as the above subsection. This examination shows that the average color differences are 3.4 for the dye sublimation type and 5.9 for the ink jet type.

Next, we have applied a matrix transformation method. In this method, the nonlinear mapping is described by a polynomial expression, and the weighting coefficients to the nonlinear terms are summarized into a matrix. This matrix is determined using the linear regression analysis based on the least-squares method. For instance, if we use the second order polynomial for the three quantities L*, a*, and b*, a ten-dimensional vector is defined as $\mathbf{x} = [L, a, b, L^2, a^2, b^2, La, Lb, ab, 1]^t$, where the symbol t denotes a matrix transposition. The transformation matrix is then obtained by solving the regression equations $[C, M, Y, K]^t = \mathbf{Mx}$, where the 4×10 matrix $\mathbf{M}$ has 40 unknown parameters. The test results using this matrix show the average color differences of 12.1 for the dye sublimation type and 12.9 for the ink jet type. Moreover we have made a large transformation matrix by permitting the fifth order polynomial of L*, a*, and b*. The color differences by this matrix are still larger than the previous errors by the proposed method.

Conclusion

This paper has described a method for realizing the mapping from the three-dimensional color space of color stimuli to the four-dimensional CMYK space of printer ink signals. The CIE-L*a*b* color system is used as the color space. Color reproduction on a printer is considered as the problem of controlling an unknown static system with four inputs and three outputs. The algorithms for the mapping are based on a two-phase procedure using neural networks, which eliminate the need of UCR and GCR.

In the first phase the unknown system of a printer is modeled with a neural network. The second phase determines the total network system combining the printer model and the controller so as to make the one-to-one mapping. Then the controller realizes the inverse mapping for the printer. The performance of the proposed mapping method was examined on experiments using a dye sublimation printer and an ink jet printer. The results show a good accuracy of color reproduction on both printers.

References

1. M. C. Stone, W. B. Cowan, and J. C. Beatty, Color gamut mapping and the printing of digital color images, *ACM Trans. on Graphics*, **7**, 249-292, 1988.
2. R. S. Berns, Color WYSIWYG: A combination of device colorimetric characterization and appearance modeling, *Proc. SID 92*, 540-552, 1992.
3. H.Kotera, et al., A single chip color processor for device independent color reproduction, *Proc. 1st Color Imaging Conf.*, 133-137, 1993.
4. Y. Arai, Y. Nakano, and T. Iga, A method of transformation from CIE-L*a*b* to CMY value by a three-layered neural network, *Proc. 1st Color Imaging Conf.*, 41-44, 1993.
5. S. Tominaga, A neural network approach to color reproduction in color printers, *Proc. 1st Color Imaging Conf.*, 173-177, 1993.
6. S. Tominaga, Color control using neural networks and its Application, *IS&T/SPIE's Sym. Electronic Imaging*, 1996
7. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning internal representations by error propagation, In D. E. Rumelhart and J. E. McClelland (Eds.), *Parallel distributed processing*, **1**, chap. 8, MIT Press, 1986.
8. S. Tominaga, Color notation conversion by neural networks, *Color Research and Application*, **18**, 253-259, 1993.