PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Color control using neural networks and its application

Shoji Tominaga

Shoji Tominaga, "Color control using neural networks and its application," Proc. SPIE 2658, Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts, (29 March 1996); doi: 10.1117/12.236974

Event: Electronic Imaging: Science and Technology, 1996, San Jose, CA, United States

Color Control Using Neural Networks and Its Application

Shoji Tominaga

Osaka Electro-Communication University
Neyagawa, Osaka 572, Japan
E-mail: shoji@tmlab.osakac.ac.jp

ABSTRACT

A method is proposed for solving the mapping problem from the three-dimensional color space to the four-dimensional CMYK space of printer ink signals by means of neural network. The CIE-L*a*b* color system is used as the color space. The color reproduction problem is considered as the problem of controlling an unknown static system with four inputs and three outputs. A controller finds the CMYK signals necessary to produce the desired L*a*b* values from a printer. Our solution method for this control is based on a two-phase procedure. Validity of our method is shown in an experiment using a dye sublimation printer.

1. INTRODUCTION

The concept of device-independent color reproduction of images has been widespread since the advent of desk-top publishing systems. Color reproduction requires color conversion between the color signals, depending on the device used, and the standard color coordinates, representing color appearance like the CIE-L*a*b* color system[1-2]. There are several methods for estimating the amounts of primary inks necessary to produce a desired color stimulus on a printer.

These are (1) an analytical method using the Neugebaouer equations, (2) matrix transformation, (3) use of a look-up table and three-dimensional interpolation, and (4) a neural network method. The first analytical method often has an inevitable discrepancy between the printer outputs and the model equations. In (2)-(4), a color printer is regarded as a black box. Method (2) describes the input-output relationship by a matrix with nonlinear elements. Method (3) makes a data table on measured color values, and interpolates this table to determine the input signal corresponding to a desired color output[3]. Method (4) models the mapping between the printer color signals and the output color stimulus values[4-5].

A neural network is suitable for modeling a nonlinear transformation between two color spaces, for which a mathematical description is difficult. This network consists of collections of connected processing units (neurons). It can learn adaptively the mapping, and store the knowledge in the network structure. The mapping from one space to another is then realized using the simple structure in which nonlinear units are linked in parallel and in layers.

Previous neural network methods were developed to control only three color signals CMY of printer primaries. However we need the four inks of CMYK for high quality color reproduction. In this case, the conversion from the color stimulus values like CIE-L*a*b* or CIE-XYZ 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. Previous network approaches fail in realizing such a mapping.

The present paper describes a method for solving the mapping problem from the three-dimensional color space of color stimuli to the higher dimensional color space of printer signals. A color printer accepts the CMYK digital values as the input, and generates the corresponding color stimulus as the output. We do not use a heuristic method for generating the black (K) ink, such as Under Color Removal

0-8194-2032-8/96/$6.00

(UCR). The CIE-L*a*b* color system is used as the device-independent color space. The mapping from the L*a*b* color space to the printer CMYK space is constructed using a neural network.

When we regard a color printer as an unknown static system with four inputs and three outputs, the color reproduction problem can be considered as the problem of controlling a system with unknown characteristics. We determine the CMYK values of the input control signals so that the printer system outputs the desired L*a*b* values. A system for the signal determination can be called the controller. The controller attempts to realize the inverse mapping of the unknown system.

2. COLOR CHARACTERISTICS OF A PRINTER

The present study uses a non-impact printer of the thermal transfer-dye sublimation type. It prints three (CMY) or four (CMYK) colors at the resolution of 300 dots per inch. Each color printer primary is expressed in 256 levels (8-bits). In the system using three colors, the two sets of signals, CMY and RGB can be converted with a simple relation for opponent color as

$$C = 255 - R, \qquad M = 255 - G, \qquad Y = 255 - B. \tag{1}$$

In the system using four colors, the K values are determined using UCR from the CMY values.

We do not use the technique of USR in the four-color printing to control the amounts of CMYK inks independently. That is, we can specify the K digital value independently, so that black ink is laid on a print after YMC. The color characteristics of our printer were investigated in the general case of four inks. Many uniform color patches were made with papers printed by changing each value of CMYK in the range [0, 255]. The object colors of these color patches were measured by a colorimeter under D65.

Figure 1 shows the relationship between the input digital value and the luminous-reflectance factor in percent Y(%) of the produced color on a paper for each single primary printing. The reflectance of Yellow primary is very high and linear with the input value, while for other primaries there is no linear relationship.

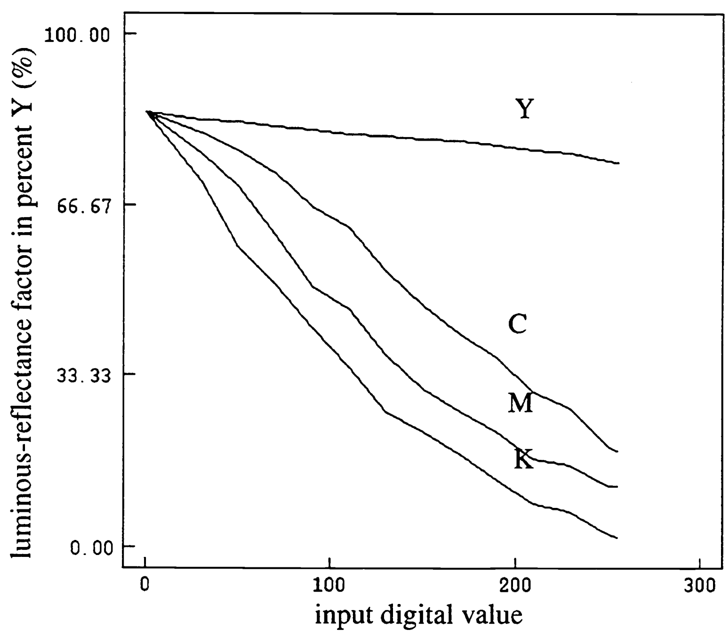

Figure 1 Relationship between the input digital value and the reflectance factor Y (%) for each primary.

Figure 2 shows the color gamut of the present printer in the CIE-L*a*b* color space. As the four dimensional input signals CMYK were changed in the full range [0, 0, 0, 0]-[255, 255, 255, 255], measurements on printer output were specified in the L*a*b* color system. The reference white standard was the printer output in [0, 0, 0, 0], that is, white of the paper. In the L*a*b* color space, we connected the outer-most coordinate points of the measured color values to create a polyhedron consisting of triangular patches. Figure 2 indicates such a surface approximation of the gamut by a polyhedron. It should be noted that the reproducible color region is not uniform. In fact, the yellow region expands into the direction of high lightness and high saturation, while the blue region is suppressed into low lightness and low saturation.

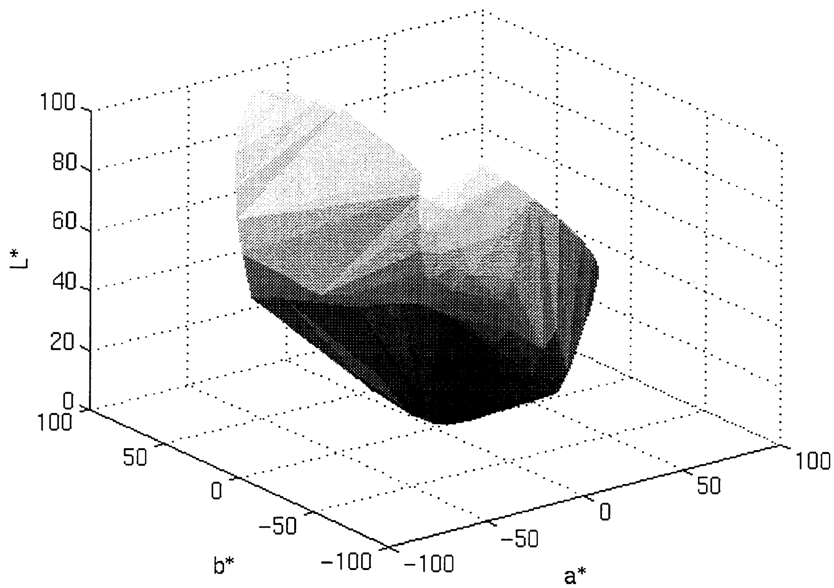

Figure 2 Color gamut of the present printer in the CIE-L*a*b* color space.

3. BASIC PRINCIPLE OF COLOR CONTROL

Color reproduction must determine the digital values of the ink signals CMYK necessary to generate any desired L*a*b* color specification within the above gamut of a printer, that is, to realize the mapping from the CIE-L*a*b* color space to the printer CMYK color space. Previous neural network methods were applied to this task, but the associated iterative learning algorithms do not converge.

We consider the color reproduction problem as the problem of controlling a system with unknown characteristics (see Figure 3). 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. We propose a method for finding this solution based on a two-phase procedure.

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. A printer model is determined by a learning procedure to minimize the color difference between a target L*a*b* value in the training data and the predicted L*a*b* value by the network.

Once the printer system is learned, in the second phase, the control system of mapping a color specification vector into an ink vector is determined using a neural network. We adopt an indirect manner to perform this. Let us consider the whole network system combining the controller and the printer (see (A) in Figure 3). 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. It should be noted that 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 second phase we determine the identity mapping for the total network system which combines the controller and the model equivalent to the printer (A). This identity mapping can be realized with a neural network. In this phase, 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.

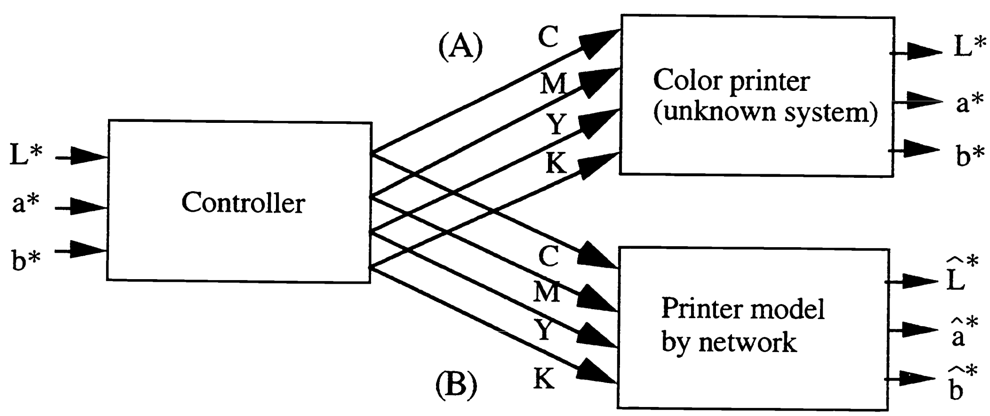

Figure 4 Color control of a printer by neural networks.

4. PRACTICAL ALGORITHMS

4.1 Determination of the printer model

Figure 4 depicts a network structure for modeling a printer. The mapping from the input space of four ink signals CMYK to the output L*a*b* space is described using a four-layered neural network. Each of two hidden layers has 10 units. In Figure 4, every unit receives its input signal from the prior layer to compute the weighted sum, and outputs a level of activation for the unit through a nonlinear function.

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 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

$$\mathrm{net} = \sum_i w_{ji} o_i + b. \tag{2}$$

The nonlinear output of unit j is

$$o = f(\text{net}), \tag{3}$$

where f is the sigmoidal activation function

$$f(\text{net}) = 1/\{1 + \exp(-4\alpha\,\text{net})\}. \tag{4}$$

This function takes any real number in the interval [0, 1], and the positive constant α represents the slope of f at net=0. The operations of (2)-(3) are executed at all units except the input layer.

All the input/output signals of the network 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] to make a cube.

If the printer model is described with a 4-10-10-3 type network as shown in Figure 4, there are 170 weighting coefficients and 23 biases. The set of these parameters are determined by a learning procedure based on the measurement data. We adopt the learning rule of error backpropagation[6].

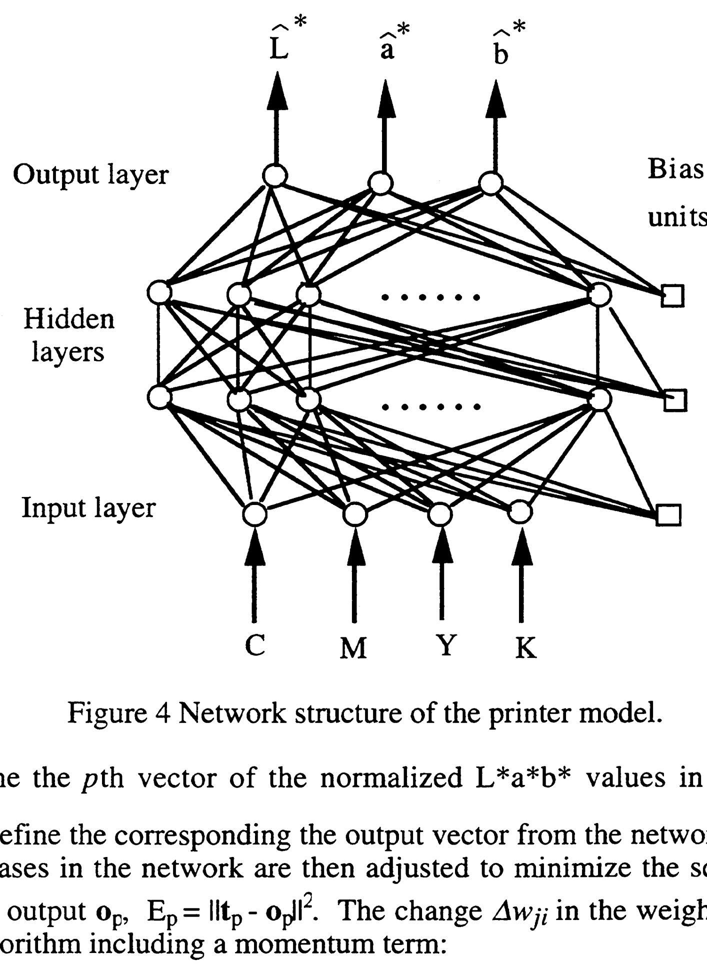

Figure 4 Network structure of the printer model.

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:

$$\Delta w_{ji}(n+1) = \eta\delta_{pj} o_{pi} + \beta\Delta w_{ji}(n), \tag{5}$$

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. 7). The notations η and β are positive constants called the learning constant and momentum constant, respectively. The change Δb_j in the bias b_j is computed in the same manner as are the other weights.

4.2 Determination of the controller

The network structure of a controller is assumed to have the same complexity as the printer model, because it should be a type of the inverse system to the printer. As shown in Figure 5, the controller is constructed with the four layered network of a 3-10-10-4 type with 3 units in the input and 4 units in the output.

First we make a large network combining the printer model and the controller as in Figure 5. This 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], compressed from the color specifications in the region of $0 \leq L^* \leq 100$, $-100 \leq a^* \leq 100$, $-100 \leq b^* \leq 100$.

The above neural network is trained to realize an identity mapping from the L*a*b* color space to the L*a*b* color space. The computations in each unit are coincident with Eqs. (2)-(4). 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. Moreover the L*a*b* values are not necessarily selected from the measured data set, but any L*a*b* values sampled in the color gamut can be used.

(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 in **4.1**, 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 5, the filled circles and squares of ● and ■ indicate, respectively, the fixed units and biases, while the units and biases of ○ and □ 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 of the network 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, accompanied with the error backpropagation.

Here we note the structure of the completed total network. The printer model part in the upper half is unchanged, while the controller part in the lower half is newly learned. Because the upper half represents the mapping from the CMYK space to the L*a*b* space, and moreover the total system performs the one-to-one mapping, the lower half must realize the inverse mapping of the printer model part, that is the mapping from the L*a*b* space to the CMYK space. In other words, the desired controller of the printer is obtained by taking out the lower half of the total network. This controller can determine the four ink signals necessary to generate any target L*a*b* color specifications on the printer.

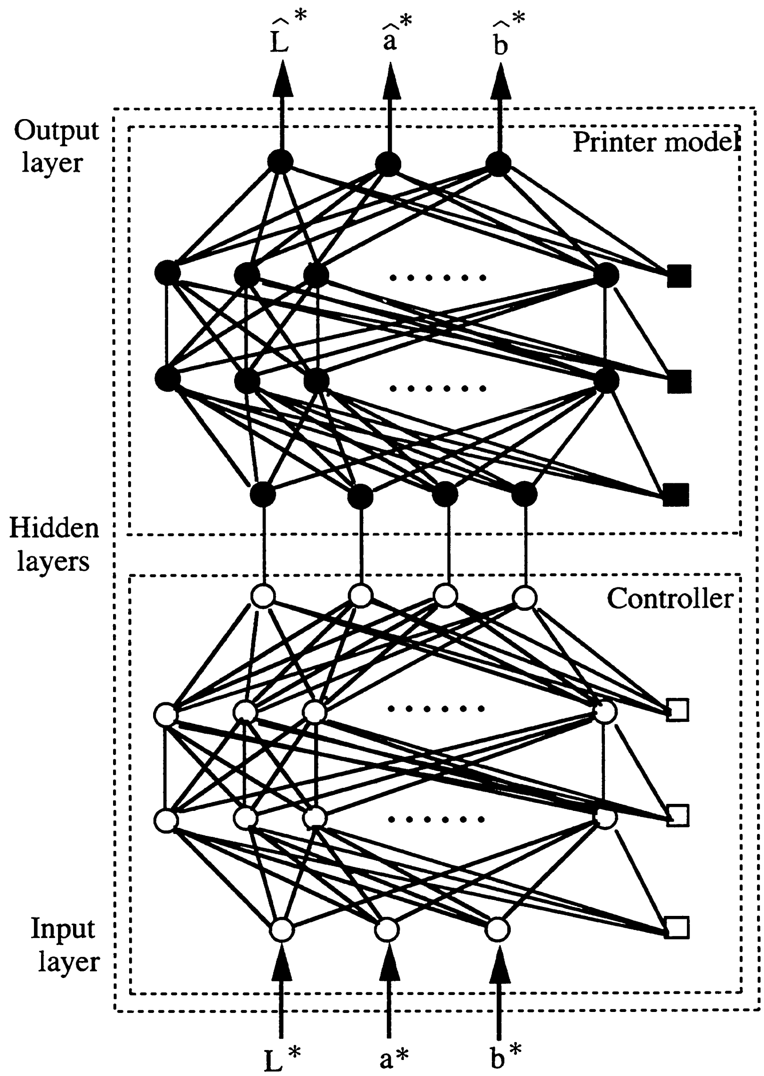

Figure 5 Network structure for determining the identity mapping and the controller.

5. EXPERIMENTAL RESULTS

First, to make the printer model, we measured many color patches printed by four inks. The total number is 8^4=4096. We obtained the training data set which consisted of 4096 pairs of the input ink signals and the corresponding L*a*b* color specifications of the printer outputs.

The networks were simulated on a SUN SPARC Station 20. The initial values $w_{ij}(0)$ and $b_j(0)$ are set to random numbers. 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 epochs of the iterative learning, the system error becomes a small value of 0.868.

Next, to make the controller, the total eight-layered network was trained according to the procedure in **4.2**. For realizing the identity mapping, we presented the same L*a*b* values to the input/output layers, which were coincident with the L*a*b* values of 4096 training data. Training the total network converges at around 40000 iterations. The controller part is then taken out from the network. The accuracy of color reproduction by the controller is examined. We used 150 samples as the test data. First, the L*a*b* color specifications of each test sample is input to the controller so that the ink amounts of CMYK are determined. Second, color patches are printed from these CMYK signals. The accuracy is examined in comparing the reproduced colors to the target colors. The mean error is about 2.5.

6. CONCLUSION

This paper has proposed a method for solving the mapping problem from the three-dimensional color space to the four-dimensional CMYK space of printer ink signals by means of neural network. The CIE-L*a*b* color system is used as the device-independent color space. We have realized the mapping from the L*a*b* color space to the printer CMYK color space. The color reproduction problem was considered as the problem of controlling an unknown static system with four inputs and three outputs. A controller finds the CMYK signals necessary to produce the desired L*a*b* values from a printer. Our solution method for this control was based on a two-phase procedure. The first phase determines a neural network modeling the printer. The second phase determines the total network system combining the printer model and the controller so as to learn an identity mapping in the L*a*b* color space. Then the network of the controller part realizes the desired mapping for color control. Practical algorithms were presented with multilayer feedforward networks. Validity of our method was shown in an experiment using a dye sublimation printer.

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, pp.249-292, 1988.
2. R.S. Berns, " Color WYSIWYG: A combination of device colorimetric characterization and appearance modeling", *Proc. SID 92*, pp.540-552, 1992.
3. H.Kotera, et al., "A single chip color processor for device independent color reproduction", *Proc. 1st Color Imaging Conf.*, pp.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.*, pp.41-44, 1993.
5. S. Tominaga, "A neural network approach to color reproduction in color printers", *Proc. 1st Color Imaging Conf.*, pp.173-177, 1993.
6. 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.
7. S. Tominaga, Color notation conversion by neural networks, *Color Research and Application*, **18**: 253-259 (1993)