Color space conversion of digital photofinishing by neural network

Baozhong Mu (穆宝忠) and Feihong Yu (余飞鸿)

State Key Laboratory of Modern Optical Instrumentation, Optical Engineering Department, Zhejiang University, Hangzhou 310027

Received December 30, 2004

A practical neural network model was designed to realize the color space conversion of digital photofinishing. The sampling, network structure and training process were introduced respectively. But in actual training, the networks fall into local minimum in all probability. To solve this problem, evolutionary programming (EP) algorithm was applied and the learning rate was adaptively adjusted. In the experiment, the performance of network was compared with pre-optimizing. Then the color space conversion was evaluated by the simulation error of samples from the point of color difference.

OCIS codes: 330.1720, 330.1690, 330.1730.

Digital photofinishing is developed with the combine of digital technology and silver printing technology. To reproduce high-quality image on printing paper, the color space conversion between colorimetric values and exposure control must be set up. However, this relation is very difficult and is almost impossible to be expressed by a theoretical equation. Currently, neural network and three-dimensional lookup table (3D LUT) are the relatively feasible method[1]. Comparatively, neural network consumes smaller system resource than 3D LUT. So back propagation (BP) neural network was used to simulate the color space conversion of digital photofinishing.

To improve the performance of network, the sample and network structure were optimized in turn. However, in actual training the local minimum still occurs in all probability. In theory, learning rate is the key factor influencing the convergence of network. But the initialized learning rate is a constant value and cannot meet the dynamic demands of weighting coefficients adjustment. Therefore, the evolutionary programming (EP) algorithm was applied, which greatly increases the probability to escape from local minimum. Once the local minimum is solved, the network comes back to the BP training. So this is a combining algorithm of BP and EP.

Color space conversion is an important step in the color management system (CMS). Through gamut mapping, the original image is converted into the profile connection space (PCS) such as CIE Lab, CIE LUV, CIE LCH etc. To get reproduction image on printing paper, the conversion from PCS to exposure control signals must be set up. The color reproduction of digital photofinishing is based on a subtractive mixture of three primaries of cyan (C), magenta (M), and yellow (Y) and each primary is stimulated by the corresponding exposure. As shown in Fig. 1, the conversion model can be divided into two steps in turn: 1) conversion from CIE Lab to CMY; 2) conversion from CMY to exposure control signals.

Usually CMY limited in the range of 0—255 is defined by the corresponding dye deposition. However, the actual values of dye deposition cannot be measured accurately and the optical density is usually used to name it. For sensitive paper, three characteristic curves are given to describe the relationship between optical density and logarithmic exposure. So CMY can be specified by corresponding exposures which are calculated by dividing the optical density of characteristic curves. So the conversion from CMY to exposure signals can be defined by three 1D LUTs which are the inverse process of CMY specifying.

For the conversion from CIE Lab to CMY, its theoretical basis is the subtractive color mixture. The spectral interaction makes this conversion quite complicated. It is almost impossible to accurately simulate this conversion by theoretical equations or a 3D matrix. In printing system, the solution is 3D LUT and neural network. Based on the consideration of sampling and system-consuming, the BP neural network was selected in our system. As shown in Fig. 1, the color conversion can be regarded as

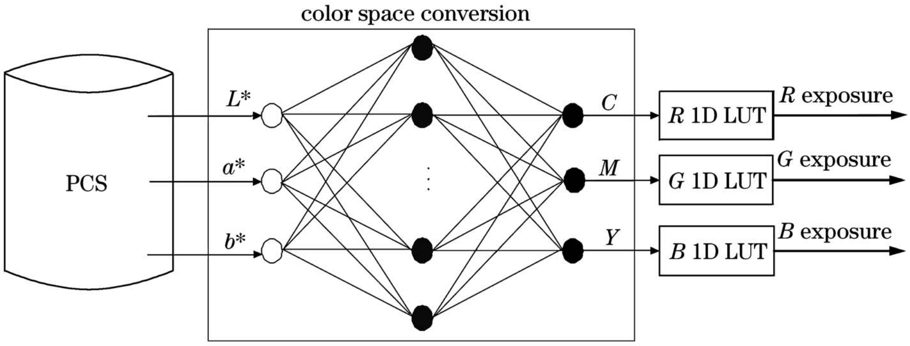

Fig. 1. Color space conversion of digital photofinishing using neural network.

static black box. Based on finite sample pairs, the black box will be simulated by the neural network. According to the order of network processing, the sampling, initializing, and training will be introduced.

The description of system by neural network comes from the samples learning, so the selection of samples will make a great impact on network. Usually, the samples should be considered from quantity and regularity. The quantity of samples should be not less than $9\times9\times9$ according to the criterion in 3D LUT. Generally, it is difficult to get quite regular samples before the color space conversion established. So we designed a slicing method to get relatively regular samples.

In this method, the color gamut was sliced uniformly by the cumulative color difference defined by

$$\Delta E_i = \sqrt{(L_i^* - L_0^*)^2 + (a_i^* - a_0^*)^2 + (b_i^* - b_0^*)^2}, \qquad (1)$$

where subscript i represents ith level of each dye and it is limited in the range of 0—255. When $i = 0$, CMYs are all equal to zero and it means the white point. By testing cyan, magenta, and yellow patches, ΔE_i curves can be simulated. As shown in Fig. 2, taking the magenta dye as an example, a single non-uniform series of slices along the horizontal axis were provided by slicing the ΔE_i curve with the settled spacing[2]. This slicing can also be extended to 3D space as well.

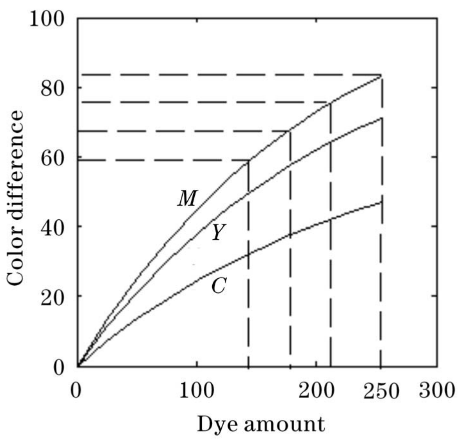

Fig. 2. Slicing uniformly in CIE lab color space with cumulative dye change.

Table 1. The Corresponding *CMY* Values for Uniform Samples in CIE Lab Space

C	M	Y
0	0	0
17	16	16
34	33	33
53	51	51
74	70	72
96	92	94
121	116	118
148	142	145
179	173	176
214	209	212
255	255	255

The spacing along vertical axis was determined by the required quantity of slices. Here, the vertical axis of each dye was sliced to 10 equal shares and the responding CMY values were listed in Table 1. Controlling C, M, and Y values simultaneous by the permutation and combination type, $11\times11\times11$ sample pairs were measured.

Now the input and output layers of network could be decided by the samples. The input layer was set at three units of L^*, a^*, and b^*, and the output layer was also set at three units of C, M, and Y. It should be noticed that all these samples must be normalized in training.

There is no effective formula to initialized structure of network. In theory, if the amount of hidden units is large enough, a BP network with one hidden layer can approach any non-linear function infinitely[3]. So the network structure could be initialized to the form of 3-N-3 where N is the size of hidden units. The value of N makes a great influence on the performance of network. If N is too small the training cannot be converged, and if N is big the over-simulation is often caused. It is difficult to get the optimum structure of network only by experience. Our solution was to generalize the performance of network with the test by changing the number of the hidden units from 10 to 30[4]. The selecting test for structure need use the aftermentioned optimizing algorithm. By comparing the global errors, the final size of hidden units was set at 30.

In BP network, each unit receives its input signals from the previous layers, computes the weighted sum, and then outputs the unit's level of activation by weighting this sum with a non-linear function. So all units except for the input layer can be described by

$$o_j = f(\sum_i \omega_{ji} o_i + b_j), \qquad (2)$$

where o_i is the output of the ith unit in the previous layer, ω_{ji} is the weighting coefficient to connect the ith unit of the previous layer with the jth unit, b_j is the bias terms, and f is the function as shown below

$$f(x) = \frac{1}{1+\exp(-x)}. \qquad (3)$$

Usually, the global system error is used to evaluate the performance of network and it is defined by

$$E = \frac{1}{2}\sum_{p=1}^{1331}(t_p - o_p)^2, \qquad (4)$$

where t_p is the normalized CMY values of 1331 samples and o_p is the output value of network. If E is larger than the given goal error, it will be returned to optimize the weighting values and bias values, and certainly the iterations will go on. In fact, the above training is only an ideal model for small size of samples. Because the sizes of samples in color space conversion are very large, the training falls into local minimum in all probability. The crucial reason is that the initialized learning rate cannot meet the dynamic demands of ω_{ji} change. To solve this problem, we had ever used adaptive learning rate defined as

$$\eta(t+1) = \eta(t)\cdot\frac{E(t)}{E(t-1)}. \qquad (5)$$

The learning rate in Eq. (5) reflects the demands of ω_{ji} change, so it would increase the probability to escape from local minimum. But the actual training result is that the training time was greatly reduced and the probability of local minimum was still high. The essential reason is that the learning rate in Eq. (5) is compulsorily adjusted. If add some random factors, the converging probability will be increased certainly.

The thinking of EP includes the consideration of random vibrations and optimizing selection[5]. So it could compensate the shortcoming of Eq. (5). Next, we will adjust the learning rate based on EP thinking.

When the training falls into the local minimum, n groups of learning rate are randomly set as η_i ($i = 1, 2, \cdots, n$). Here η_i is regarded as the superset of EP process. Applying η_i to the current training, n groups of global system errors E_i can be calculated by Eq. (4).

We define S_i as the adaptive ability of η_i through

$$S_i = 1/\exp(E_i - E_{\min}), \quad (i = 1, 2, \cdots, n). \tag{6}$$

The evolving factor σ_i is defined by

$$\sigma_i = \sqrt{\exp(E_i - E_{\min}) - 1 + 10^{-6}}, \quad (i = 1, 2, \cdots, n). \tag{7}$$

Using the evolving factor, the subset learning rate is generated by

$$\eta_i' = \eta_i + \sigma_i \cdot N_i(0, 1), \tag{8}$$

where $N_i(0, 1)$ is a random value limited in the range from 0 to 1 and it is various for different η_i. To express conveniently, we name the aggregation of superset and subset learning rates with V. The unit of V is named with v_i ($i = 1, 2, \cdots, 2n$). The next generation will be evolved by V. Firstly, q groups of units are randomly selected from aggregation V and form the new aggregation expressed with ϕ. Comparing unit v_i with ϕ, we can count such units in ϕ whose adaptive ability coefficients are smaller than those of v_i. The results are expressed with W_i. Here $W_i \leq q$ and $i = 0, 1, \cdots, 2n$. Then n groups of relatively bigger W_i are selected and their corresponding learning rates are inherited. The inherited learning rates mean relatively stronger viability and they will serve as the new superset in next EP process. When $E_{\min} \leq 0.9E_{\mathrm{local}}$, the local minimum can be thought solved. So the evolution should be stopped and the BP training will be continued. The whole training process is an actual combined algorithm of BP and EP.

To control the training time, the total evolution times G need be set. When G is achieved, it means the EP cannot solve the current local minimum and the training will be stopped.

In the experiment, the initialized values were the same for all networks except for the selection of random value. The size of hidden units was set 30, the total iterative times was set epochs $= 1 \times 10^7$, the initialized learning rate $\eta = 0.05$, the goal performance $\varepsilon = 0.01$. The global error was used as the evaluation of network performance. In tests using common BP network, the best global error is $E = 3.55$, most of global errors are concentrated around 9. Obviously, common BP network cannot be converged with current network settings.

Table 2. Analysis of Simulation Errors by the Optimum Network

Error	$\lvert\Delta C\rvert$	$\lvert\Delta M\rvert$	$\lvert\Delta Y\rvert$
Mean Error	1.53	1.62	1.51
Maximum Error	4.31	4.40	3.72

In tests with BP-EP network, the size of each generation was set $n = 30$, the number of election in EP was set $q = 18$, and the total times of EP was set $G = 200$. The result is that nearly one third networks could be converged and the global errors are all less than 0.6. The best result is $E = 0.2448$. For the other networks, the global errors are mainly distributed in the range from 3 to 5 and all these networks stopped training in the middle. Certainly, if set a larger G value, the issue may be improved in a certain extend. According to the experiments, we can see though the combine algorithm the performance of network improved greatly, the selection based on random experiments is also needed.

To evaluate the feasibility of the color space conversion using BP-EP network, the samples were simulated by $E = 0.2448$ network. The mean absolute errors and maximum errors were listed in Table 2.

The error need be evaluated from the point of color difference of digital photofinishing. In the different regions of gamut, color differences caused by constant CMY error are also different. Through the grey test, we found that the caused color difference increased with the decrease of luminance. If set the CMY error ΔC, ΔM, $\Delta Y = 5$, nearby the black point of printing paper the caused color difference is $2.8\Delta E_{\mathrm{ab}}$, and nearby the white point it is $0.9\Delta E_{\mathrm{ab}}$. By experience, two images viewed side by side with systematic errors that result in an average ΔE^*_{ab} of 2.2 or less are indistinguishable from one another[6]. If the color differences achieve $5\Delta E^*_{\mathrm{ab}}$ on average, the accuracy is acceptable for high-quality color reproduction. Obviously, the simulation error is close to the indistinguishable color difference level. Even if the systematical error is considered, this conversion can also achieve the color difference demands of high-quality color reproduction.

This work was supported by the National High and New Technology Industrialization Foundation of China (No. 2001AA300152). B. Mu's email address is mubaozhong1975@sina.com.

References

1. H. R. Kang, Proc. SPIE **2413,** 410 (1995).
2. K. D. Gentennet, Proc. SPIE **1909,** 117 (1993).
3. G. Cyberko, Math Control Signal System **2,** 303 (1989).
4. B. Chai, N. Liao, and D. Zhao, Chin. Opt. Lett. **2,** 637 (2004).
5. D. B. Fogel, IEEE Trans. Neural Network **5,** 3 (1994).
6. R. S. Berns, IS&T **45,** 305 (2001).