A colour printer calibration method based on gamut division algorithms

Bangyong Sun, a, * Han Liu, b Shisheng Zhou, a Junhong Xing, c Congjun Cao a and Yuanlin Zheng a

a School of Printing and Packaging, Xi’an University of Technology, 5# Jin Hua Road, Xi’an, Shaanxi, China
Email: sunbangyong@xaut.edu.cn
b School of Automation and Information Engineering, Xi’an University of Technology, 5# Jin Hua Road, Xi’an, Shaanxi, China

Received: 12 October 2013; Accepted: 4 September 2014

A colour printer calibration method based on gamut division algorithms is proposed in this paper, with the main task of converting colours from CIELab coordinates to CMYK space. The printer 3D gamut is firstly divided into seven subgamuts according to the K values of sample data, and then the subgamuts, which enclose the given point in CIELab space, are found. Because the K value stays the same for all sample colours of the determined subgamuts, a polynomial regression method is employed to convert the colours from CIELab space to CMY. Finally, by using black generation coefficients, the complete CMYKs are obtained according to the image characteristics. In experiments, for 256 test colours, the mean colour error of our algorithm, measured as $\Delta E_{2000}^{*}$, is 1.45, which is acceptable for most printing products. Furthermore, the errors are less than those in both Neugebauer equations and the neural network method. These positive features suggest that the precision of our method is acceptable.

Introduction

Colour spaces can be divided into two types: colourant spaces (or device-dependent spaces, such as RGB or CMYK spaces) and colorimetric spaces (or device-independent spaces, such as CIEXYZ or CIELab spaces). Via different colour devices, even the same colourant values often produce different visual senses. Thus, colorimetric spaces were used as the standard colour spaces to link different devices during colour transmission. Because the colourant space within printers is CMYK, the input signals, which are often colorimetric colours, need be converted into CMYKs before printing. Hence, for the printer calibration algorithms (or separation algorithms), the main task is to obtain the precise relationship between these two types of colour space. The ideal calibration algorithm should minimise the visual difference between the input and the output colours, and by comparing the colorimetric errors the algorithm can be evaluated.

For CMYK printers, black colour (K) is used to replace certain combinations of CMY colours, which explains why many different CMYK values reproduce the same visual effect. Printer calibration algorithms are aimed at converting the given colorimetric colours into CMYKs without changing the visual effect. These algorithms can be classified into two groups as follows.

In the first group, colorimetric colours are firstly converted into CMY, and then the K is created from CMY by using black generation algorithms, such as the UCR (Under Colour Removal), GCR (Grey Component Replacement), and UCA (Under Colour Addition) algorithms. Thus, these CMYK separation processes can be divided into two steps: colorimetric to CMY and CMY to CMYK. Because the gamut of CMY is smaller than CMYK for a four-colour printer, this separation process does not take full advantage of the printer’s gamut.

In the second group of calibration methods, different sequences are used to convert colorimetric colours into CMYK for the purpose of utilising the entire range of reproduction capability. For example, Ogatsu et al. [1] introduced a flexible GCR algorithm from CIELab to CMYK, where K is determined ahead of CMY by binary search, the K adjustment coefficient is controlled by two defined points and the chroma values, and the CMY conversion is based on 3D-LUT data. Tsukada and Tajima have proposed an algorithm for UCR [2], using the direct colour mapping method from RGB to CMYK, where the conversion from RGB to CMY is based on tetrahedron interpolation [3], which requires a great amount of sample colours to guarantee conversion precision. Thus, it must store a large amount of sample data, which limits computational efficiency. Zeng [4] has developed a new GCR algorithm by directly converting CIEXYZ values to CMYK values. In this algorithm, the K amount can be flexibly controlled by users, but the precision is highly influenced by the additivity of colorimetric density. There are also some other models that can be used in direct CMYK conversion algorithms, such as 4D-LUT [5], the neural network method [6], or the Neugebauer model [7]. Although these methods can quickly calculate the CMYK from given colorimetric colours, there are some problems that limit the application of these algorithms; in particular, the obtained K value cannot be flexibly adjusted.

In this paper, a flexible CMYK separation algorithm is proposed that is based on gamut division so as to make the K value adjustable for different images. The printer gamut is firstly divided into seven subgamuts according to the K values of the sample data, and then for the given colour in CIELab space its K value is determined by comparing all the subgamuts that enclose it, so that the CMY colours can be obtained by using three-ink conversion methods.

Printer Gamut Division Method

For a CMYK printer, its colour gamut can be obtained by using the gamut description method with all the sample
data. As the K value has a great influence on gamut appearance, the printer gamut is divided into several subgamuts according to K in order to analyse that influence. By using the sample data, which include CMYK and the corresponding colorimetric values (CIELab colour space in this paper), the gamut is divided into seven subgamuts, where the K values are 0, 10, 20, 40, 60, 80, and 100 respectively. When the K remains the same in each of the seven subgamuts, the C, M, and Y values change gradually. In Figure 1, as the K value gets larger, the corresponding subgamut moves down along the CIE L* axis, and so the volume becomes smaller. This means that the colour range decreases as the K value increases.

As shown in Figure 1, although the subgamuts vary greatly for different K values, they have some obvious overlapping regions. Thus, a given colour in CIELab space may be contained by more than one subgamut, which means that a colorimetric colour often corresponds to several CMYK colours. Therefore, if a colorimetric colour is enclosed by merely one subgamut, then the colour conversion from CIELab to CMYK is carried out in that subgamut. When CMYK separation is carried out in a subgamut, all the related sample data have the same K value, this turns into a CMY three-ink separation process. When two or more subgamuts enclose a colorimetric colour, there are several CMYKs obtained after conversion in these subgamuts, and the final CMYK should be a combination of these CMYKs with black generation coefficients.

In the CIELab-to-CMYK separation algorithm proposed above, the given colour in CIELab space is primarily checked to decide which subgamuts enclose it. However, there may be a case where no subgamut encloses the given colour, which means it is outside the printer gamut. If so, the user-defined gamut mapping algorithm should be employed to map it into the printer gamut. Hence, after the gamut mapping process, any colorimetric colour can be obtained in the manner described above. The proposed algorithm can be described using the following pseudocode:

```plaintext
lab_in=gamut_mapping(gamut_all, lab) // map the given colour into the printer gamut
K_max=0;
K_min=100;
for (i=0;i<7;i++)
{
    flag(i)=gamut_checking(sub=gamut[i], lab_in);
    If (flag(i)=1)//the given colour is enclosed by the ith subgamut
    {if (K(i)>1) K_max
        K_max=K(i);
        if (K(i)<1) K_min
        K_min=K(i);
    }
}
CMYK_max=lab2cmy(gamut_k_max, lab_in);// three-ink conversion algorithm
CMYK_min=lab2cmy(gamut_k_min, lab_in);
CMYK=bg*CMYK_max+(1-bg)*CMYK_min
```

The description above relates two key algorithms: gamut checking and lab2cmy three-ink conversion. The former algorithm detects which subgamut encloses the given colour and can be reckoned as part of the gamut mapping process; the latter conversion algorithm is used to calculate the given colour CMY values.

**Gamut Mapping Algorithms**

For colour printers, gamut mapping is the technique of replacing non-printable colours with printable ones. It contains two steps: gamut boundary calculation and the colour mapping process. Dozens of mapping algorithms have been developed by researchers or organisations, and they can be distinguished according to the rendering intent of the images. In the ICC specification, four rendering intents are defined for users, which are perceptual, saturation, relative colorimetric, and absolute colorimetric [8]. For the purpose of evaluating the error of the calibration algorithm, the absolute colorimetric rendering intent is used, which leaves colours that fall inside the printer gamut unchanged while out-of-gamut colours are clipped. To determine whether a colour falls inside the gamut, the 3D and 2D gamut boundaries should firstly be calculated accurately. Nowadays there are many gamut boundary calculation algorithms, which are often called gamut boundary description algorithms (GBDs). According to Bakke's evaluation for the widely used GBDs [9], the precision of the SMGBD algorithm and convex hull algorithm are relatively high. In this paper, the SMGBD algorithm is used to find the gamut boundary points from sample data, and then the 3D gamut is described with the modified convex hull algorithm.

Owing to its good computational efficiency and high precision, the SMGBD algorithm is recommended by the CIE [10]. In the algorithm, a grey point is firstly appointed as the focus centre with the CIELab coordinates [L_c, a_c, b_c] = [50, 0, 0]. Then, a sample colour in CIELab coordinates is converted into a polar colour with polar vector r and two polar angles h, 0:
Figure 1 The subgamuts with different $K$ values [Colour figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]
Set in the polar coordinates, the CIELab space can be divided into \( n^2 \) sectors, and all the sample colours are mapped to the corresponding sectors. Among all the sample colours in every sector, the colour with the maximal polar vector \( r_{\text{max}} \) is selected as the 3D gamut boundary point, while the 3D and 2D gamut descriptions are all based on these boundary points. However, owing to the limited number of sample colours and the gamut characteristics of the colour device itself, some sectors may have no boundary points or may misidentify an in-gamut point as a boundary point. In that case, the gamut boundary has poor smoothness, which often causes errors in gamut mapping algorithms.

Figure 2 gives the results of gamut mapping for an IBM (USA) T23 monitor and an Epson (Japan) 9880 printer. The two gamut boundaries are different in smoothness. The professional Epson printer has a relatively smooth gamut boundary obtained by the SMGBD algorithm, whereas the boundary of the ordinary IBM monitor has some obvious uneven intervals (concave points) obtained by the same algorithm.

To eliminate the influence of the inaccurate boundary points, they are removed from their sectors, and the 3D gamut is described using the modified convex hull algorithm with the remaining boundary points. The modified convex hull algorithm, introduced by Balasubramian and Dalal [11], aims to find a relatively accurate gamut boundary consistent with the real printer gamut. To compensate for the effect of concave points, their positions are adjusted before gamut description. In this algorithm, the updated new point \( \bar{p} \) is calculated from the original position \( p \) using a gamma function based on the distance to the centre point \( e \) and parameter \( \gamma \):

\[
\begin{align*}
    h &= \tan^{-1} \frac{b^* - b_c^*}{a^* - a_c^*} \\
    \theta &= \tan^{-1} \frac{L^* - L_c^*}{\sqrt{(a^* - a_c^*)^2 + (b^* - b_c^*)^2}} \\
    r &= \sqrt{(a^* - a_c^*)^2 + (b^* - b_c^*)^2 + (L^* - L_c^*)^2}
\end{align*}
\]

Parameter \( \gamma \) controls the amount of inflation. It is often determined by trial and error and is set between 0 and 1; \( \gamma = 1 \) does not alter the position of a point, which is equivalent to a standard convex hull. As \( \gamma \) decreases, the amount of inflation increases. In particular, as the value of \( \gamma \) approaches 0, the position of the corresponding colour becomes closer to the surface of a sphere. As a result, points close to the convex surface are mistaken as being on the surface. A continuous and closed 3D gamut can be described by the modified convex hull method using the discrete gamut boundary points obtained by the SMGBD algorithm.

As most of the gamut mapping algorithms are executed in the 2D gamut boundary, which is often defined as the line gamut boundary (LGB), the LGB should be separated from the 3D gamut before the gamut mapping process. The constant-hue LGB, for example, can be deemed to be the intersection line of the 3D gamut and the constant-hue plane. Using the obtained 2D gamut boundaries, it is established whether a given colorimetric colour falls inside the gamut, and then, if not inside, it is mapped by the gamut mapping algorithm. In this paper, the clipping gamut mapping algorithm is used, which keeps the in-gamut point unchanged and maps the out-of-gamut point on the boundary line.

### CMYK Four-Colour Conversion Process

#### Polynomial regression method

The main task of the printer calibration process is to compute CMYKs from the given colorimetric colours. As before, sample data are divided into subgamuts according to the \( K \) values. When a subgamut enclosing a given colour in CIELab space is found, the \( K \) value of CMYK is determined, and the remaining CMY values can be obtained by using three-ink conversion methods. The polynomial regression method is selected in this paper as the colour conversion algorithm [12,13], as it obtains high accuracy with less sample data compared with the 3D interpolation method.

The general form of an \( n \)th-degree polynomial with three variables can be expressed as follows:

\[
f(x, y, z) = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=0}^{n} a_{ijk} x^i y^j z^k
\]

where \( x \) represents the polynomial coefficients, \( n \) is the degree of the polynomial, and \( i + j + k \leq n \). Thus, the first-, second-, and third-degree polynomials can be expressed as follows:

\[
\begin{align*}
    f_1(x, y, z) &= a_0 + a_1 x + a_2 y + a_3 z \\
    f_2(x, y, z) &= a_0 + a_1 x + a_2 y + a_3 z + a_4 xy + a_5 xz + a_6 yz \\
    f_3(x, y, z) &= a_0 + a_1 x + a_2 y + a_3 z + a_4 xy + a_5 xz + a_6 yz + a_7 x^2 + a_8 y^2 + a_9 z^2 + a_{10} x y z + a_{11} x^2 y + a_{12} x y^2 + a_{13} x^2 z + a_{14} x z^2 + a_{15} y^2 z + a_{16} y z^2 + a_{17} x^3 + a_{18} y^3 + a_{19} z^3
\end{align*}
\]

If \( c \) represents the colourant colour CMY, then the colour \( t \) in CIELab space is expressed as

\[
\begin{align*}
    \tilde{p} &= |\tilde{p} - \hat{e}| \\
    \bar{p} &= |\tilde{p} - \hat{e}| + \hat{e}
\end{align*}
\]
where $F_{\text{device}}(\cdot)$ represents the non-linear relationship between CMY and CIELab, and $\Omega_{\text{device}}$ is the colourant dataset. Similarly, if $G_{\text{device}}$ is the colorimetric dataset, then the inverse conversion process from CIELab to colourant colours can be expressed as

$$c = F^{-1}_{\text{device}}(t), \quad t \in G_{\text{device}}$$

In the calculation of the inverse conversion coefficients, the sample data containing the CMY and CIELab values are firstly extracted. If the CMY colour is expressed as $c_i = [C_i, M_i, Y_i]^T$, $1 \leq i \leq N$, where $N$ is the number of sample colours, then the CMY colour set $G_i$ is an $N \times 3$ matrix which can be shown as

$$C_i = [c_1, \ldots, c_N]^T$$

Similarly, the colorimetric colour set $T_s$ is expressed as

$$T_s = [t_1, \ldots, t_n]^T$$

where $t_i = [L_i, a_i, b_i]^T$ are the discrete colours in CIELab space. If the coefficient matrix is represented as $M$, the relationship between $C_i$ and $T_s$ is

$$C_i = T_s M$$

When the matrix $M$ is solved, it should satisfy the relationship

$$M = \arg \left( \min_{i=1}^{k} ||t_i M - c_i||^2 \right)$$

According to the least-squares method, the matrix $M$ should be

$$M = (T_s^T T_s)^{-1} T_s^T C_i$$

Finally, the inverse calibration model $F^{-1}_{\text{device}}(t)$ can be expressed as

$$F^{-1}_{\text{device}}(t) = (T_s^T T_s)^{-1} T_s^T C_i$$

### Final output CMYK value calculation

The above analysis indicates that, if the given colorimetric colour falls in only one subgamut, the $K$ value is the same as all the sample data in the subgamut, and then the CMY can be calculated by the polynomial regression method. However, there are often several subgamuts enclosing the given colorimetric colour, so the black generation coefficient should be set to make the $K$ value changeable. In the present work, when two or more subgamuts enclose the given colour, the two subgamuts with maximal and minimal $K$ values are selected, and the final CMYK is determined by using the black generation coefficient as follows:

$$\begin{pmatrix} C \\ M \\ Y \\ K \end{pmatrix} = bg \cdot \begin{pmatrix} C_M \\ M_M \\ Y_M \\ K_M \end{pmatrix} + (1 - bg) \cdot \begin{pmatrix} C_m \\ M_m \\ Y_m \\ K_m \end{pmatrix}$$

where $bg$ is the black generation coefficient, $0 \leq bg \leq 1$, and $C_m M_m Y_m K_m$ and $C_M M_M Y_M K_M$ are the maximal and minimal CMYKs converted in their corresponding subgamuts.

### Testing of Algorithm Performance and Comparison with Other Methods

Firstly, a set of samples was created for training purposes. An Epson 9880C inkjet printer was calibrated to print out colour patches (10 × 10 mm square) from known CMYK values onto 120 g m$^{-2}$ coated paper. The CIELab coordinates of each colour patch was determined using an X-Rite (USA) 528 spectrodensitometer under the following conditions: 45°/0° measurement geometry, 3.4 mm spot size at sample, D50 illuminant, and $10^\circ$ standard observers. The training sample is mainly used for obtaining the relationship between CIELab and CMYK, and it consists of seven parts with different $K$ values. Within each part of the training sample, the $K$ value stays the same, while the $C$, $M$, and $Y$ values change in sequence. It should be noted that the division of the training sample is not entirely fixed. Actually, the subdivisions were chosen in an arbitrary manner. We made divisions within the boundary conditions $K = 0$ and $K = 0$, giving six subdivisions. To accommodate the need for greater differentiation in the region $K = 0$ and $K = 20$, we added another division at $K = 10$, giving a total of seven subdivisions. This number is a compromise. An increase in the number of subdivisions improves accuracy but increases the time of measurement. Similarly to the training sample, the test sample was printed out for algorithm performance evaluation, which consisted of 256 random distribution colour patches.

Three colour separation algorithms were tested in the experiment: the proposed new algorithm based on gamut

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Mean error</th>
<th>Maximal error</th>
<th>Minimal error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neugebauer model</td>
<td>2.34</td>
<td>7.33</td>
<td>0.34</td>
</tr>
<tr>
<td>Neural network</td>
<td>1.87</td>
<td>5.48</td>
<td>0.12</td>
</tr>
<tr>
<td>Our algorithm</td>
<td>1.45</td>
<td>2.30</td>
<td>0.02</td>
</tr>
</tbody>
</table>

![Figure 3](image-url) The colour difference histogram for all test patches [Colour figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]
division, the four-colour Neugebauer model [14], and the neural network method [15,16]. Within the latter two separation algorithms, the CIELab values are directly converted into CMYK without gamut checking. The Neugebauer model is usually applied to predict the reflection spectrum from CMYK values. In the present work, its inverse model was used for colour separation as described by Gerhardt and Hardeberg [14]. The RBF neural network with three input variables and four output variables was employed to calculate CMYK from CIELab. The RBF structure and parameters were set as in Cao and Liu [16].

For each colour separation algorithm, the initial CIELab value \( L_1^* a_1^* b_1^* \) is firstly converted into CMYK values by using colour separation algorithms, and then the calculated CMYK is printed and a new CIELab value \( L_2^* a_2^* b_2^* \) is measured. Thus, the accuracy of the colour separation algorithm can be evaluated by comparing the difference between \( L_1^* a_1^* b_1^* \) and \( L_2^* a_2^* b_2^* \). In this study, the colour difference formula CIE\( \Delta E_{76} \) [17,18] was used:

\[
\Delta E_{ab}^* = \sqrt{\left( L_1^* - L_2^* \right)^2 + \left( a_1^* - a_2^* \right)^2 + \left( b_1^* - b_2^* \right)^2}
\] (14)

Finally, the errors of the three colour separation algorithms are listed in Table 1. Within the new proposed algorithm, the mean and maximal colour difference, measured as \( \Delta E_{ab}^* \), is 1.45 and 2.50 respectively. The histogram of the error is described in Figure 3, which shows that most of the test sample colour difference is less than 2.0. In the printing industry, the mean colour difference is often required to be less than 5.0, so the calibration algorithm in this paper meets the requirement for most printing situations [19]. For the other two methods, the maximal errors both exceed 5.0, and many larger errors exist in the dark tone. On the whole, the mean and maximal errors of our algorithm are significantly smaller than those of the other two algorithms.

Conclusions

Cyan, magenta, yellow, and black are the primary colours for most printing devices, and even in multi-ink outputting systems the CMYK inks are most frequently used because of their advantage economically. Therefore, it is very important to improve the accuracy of the CMYK separation process. In this paper, a new CMYK separation algorithm based on gamut division is proposed, which actually treats the four-colour separation process as a combination of several three-colour separation processes. Because \( K \) can be replaced by a certain amount of \( C, M, \) and \( Y \), one CIELab value often corresponds to several CMYK combinations, and CMYK separation is more difficult to control than CMY separation. We compared the results of using our model with the results of using the four-colour Neugebauer model and the RBF neural network method. Our algorithm gave the lowest mean error of the three methods, especially for dark tones. In addition, only one CMYK is generated for one colour represented by CIELab coordinates within these two methods; that is to say, the \( K \) value cannot be freely adjusted. The main value of the new proposed algorithm is that it simplifies the four-colour separation process with gamut division, and the CIELab-to-CMYK process is changed into several CIELab-to-CMY processes within the corresponding subgamuts. Thus, many simple and accurate three-colour conversion models may be applied to resolve the complicated four-colour separation problem, and the method based on gamut division may also be extended to the multi-ink colour separation process.

Acknowledgements

This work is supported by the National Science Foundation of China (No. 61174101 and No. 61205135), the China Scholarship Council, China Postdoctoral Science Foundation (no. 2014M562436), and the Doctor Foundation of Xi’an University of Technology (No. 104-211302).

References
