A Neutral Axis Calibration Method Using 3D Inverse Interpolation

Yifeng Wu
Hewlett Packard Company
Vancouver, Washington

Abstract

Human visual system is very sensitive to the color variation near the neutral axis. This makes the neutral axis calibration critical for color printers. For a photo printer, the neutral axis calibration is even more important, because people often use color printer to print black and white photos.

This paper presents a neutral axis calibration method based on 3D inverse interpolation. We show at first how to define the neutral axis, a CIELAB goal is defined for each RGB input located in the neutral axis. Then we describe how to design a color target and how to select near neutral colors for the purpose of measurement. Finally, we present an inverse 3D interpolation algorithm which can calibrate very accurately the neutral colors for a printer.

Introduction

The neutral axis calibration is an important step for color printer characterization and calibration. Psychological experiments showed that human visual system is more sensible to the variation of the near neutral colors. This phenomenon is considered in the CIE color difference equation published in 1994[1]:

$$\Delta E_{94} = \left[\left(\frac{\Delta L^*}{k_L S_L} \right)^2 + \left(\frac{\Delta C^*_{ab}}{k_C S_C} \right)^2 + \left(\frac{\Delta H^*_{ab}}{k_H S_H} \right)^2 \right]^{\frac{1}{2}}$$

with:

$$S_L = 1$$

$$S_C = 1 + 0.045 C^*_{ab}$$

$$S_H = 1 + 0.015 C^*_{ab}$$

where L*, C* and H* represent lightness, chroma and hue respectively. This formula shows that with the same ΔL^*, ΔC^* and ΔH^*, ΔE_{94} will be increased when chroma is decreased.

For a color printer, the neutral balance represents the balance of the total color system. In commercial printing, the neutral balance is often the first work to do before starting the printing jobs. It is even more critical for a photo printer, because people use often a color printer to print black and white photos.

The neutral axis adjustment is required in two different cases: color characterization and color calibration. Color characterization is a process to define the color rendering behavior of a color device. A 3D color map or an ICC profile is generated after the characterization. It is often performed offline. Color calibration[2-4] is a process to compensate for the color variation when printing condition is changed. Its purpose is to ensure the consistency of the color reproduction. It should be frequently executed at the run-time. The color calibration may be applied to a small number of colors[5] (e.g. neutral colors), because the color rendering accuracy and the color consistency of those colors may be more important than the others. The neutral axis calibration method proposed in this paper can be applied to both color characterization and color calibration. However we have paid more attention to color calibration, because it is an online process, the measuring data is noisier, the computational time is more critical and the algorithm should be more robust.

There are two classical ways to adjust the neutral axis. The first one is the manual adjustment, people print some neutral colors, and then manually modify the quantity of each ink component to get approach to the neutral. This procedure is repeated until the result is satisfactory. It is a tedious work, and the final result depends largely on the operator. The second method is automatic: the neutral axis is adjusted when generating the color map or ICC profile. In this case, the color patches used cover the total color space, and its number is often limited (typically several hundreds), there are not sufficient neutral and near neutral colors to calibrate accurately the neutral axis.

We propose an automatic neutral axis calibration method[6] in this paper. We show at first how to define the neutral axis: the impact of the paper white to the human preference of the neutral colors is discussed, and a CIELAB goal is defined for each RGB input located in the neutral axis. Then we describe how to design a color target and how to select near neutral colors for the purpose of measurement. Finally, we present a two-step 3D interpolation algorithm to calibrate the neutral axis in an existing 3D color map.

Definition of the Neutral Axis

The neutral axis, also called the gray axis, is the axis that links the black point to the white point. It includes all achromatic colors in the total color space. Figure 1 shows a neutral axis in RGB color space. Where the neutral colors are generally defined by the equal amount of RGB (R=G=B).

In CIELAB color space, the neutral colors are theoretically defined by a*=0, b*=0. But we should rethink this definition for a real color printing system. A color printer cannot reproduce exactly the output colors as we desired due to the inherent rendering errors, the noise, and the impact of the environmental conditions. When we define a target goal of a* and b*, we must tolerate some variation around it.

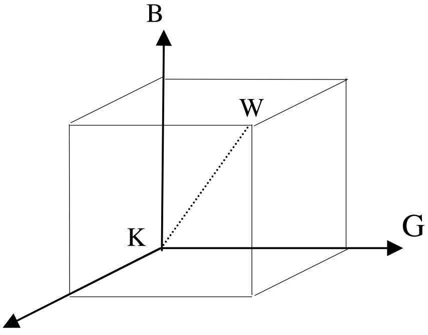

Figure 1. The neutral Axis (in dashed line)

Figure 2 describes two different definitions of the neutral axis. The first one is a*=0 and b*=0. With this definition, a positive deviation (e.g. +1) will produce a reddish gray, and a negative deviation (e.g. -1) will produce a greenish gray. If we print a neutral ramp from 0 to 255, we will have several transitions from reddish gray to greenish gray. Since the human eye is very sensitive to the neutral colors, those transitions can produce highly undesirable visual effects. In the second case, we define the neutral axis as a*=1 and b*=-3, a same color variation from -1 to +1 will always produce reddish grays. And this kind of color variations is much less visible than the transition from greenish gray to reddish gray. In other words, if we define a*≠0 and b*≠0, it is a little bit less neutral, but it produces much more consistent result at the presence of the noise. That's the reason why we prefer to define the neutral axis close to the origin of a*-b* plane, but not at the origin of a*-b* plane.

The problem remains is how to choose a* and b* value for the neutral axis? Bala[7] has presented some psychophysical experimental results about the human preference of the gray colors. He showed that the preferred gray in reflection prints occurs in a region in the chrominance plane which lies predominantly in the quadrant where both a* and b* are negative. He showed also that the gray preference depends on the illuminant. Under D50, both a* and b* are negative, but under illuminant A, a* moves toward the positive area.

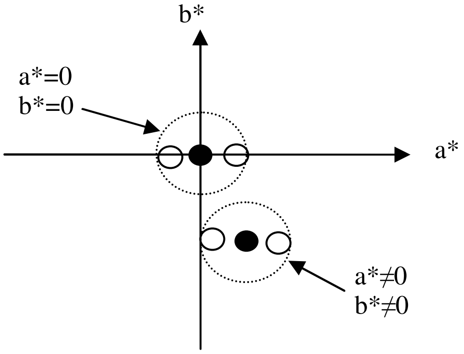

Figure 2. Two definitions of the neutral axis: a= 0, b*= 0: reddish to greenish transition*

Our psychophysical experiment got similar results. The only difference is: the human preference depends not only on the illuminant, but also on the white point of the media. The paper white of the media we used has a chrominance value of a*=0.8, b* = -3.5 under illuminant D50. In the first test, we printed 4 near neutral ramps with a*=1, and b*=0, -2,-4,-6 respectively. 6 people participated this test, 5 of them preferred a*=1 and b*=-4, and 1 preferred a*=1, b*=-2 as the most neutral color. In the second test, 4 near neutral ramps were printed with b*=-4, and a*=-1,0,1,2 respectively. 8 people participated this test, 7 of them preferred a*=1 and b*=-4, and 1 preferred a*=0, b*=-4 as the most neutral color. These two tests show that people prefer to select the neutral axis near the paper white. The dependency of the neutral preference to the paper white is more evident for light grays, if the neutral colors and the paper white have difference chrominance values, and if the light grays are surrounded by the paper white, these grays may appear colorful. In this application, the paper white is in the quadrant of a*>0 and b*<0, we define therefore the neutral axis in this quadrant of the chrominance plane. This is another reason why we prefer to define the neutral axis as a*≠0 and b*≠0. If for any reason the neutral axis you defined is different from the paper white, an interpolation should be done from a certain gray level to ensure a smooth transition from the neutral color to the paper white. And so forth for the black point.

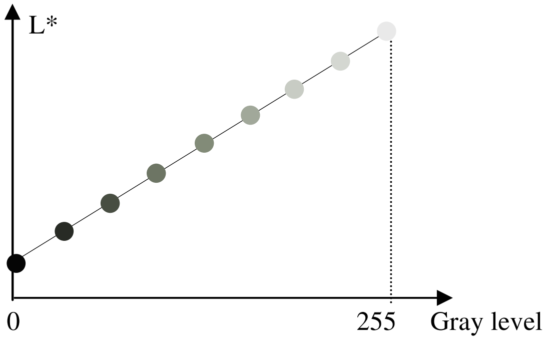

Figure 3. L is a linear function of the input gray level.*

The definition of the L* goal of the neutral axis determines the tone reproduction characteristics of the printer. We choose L* to be a linear function of the input gray level, knowing that in the neutral axis we have always R=G=B. Figure 3 shows such a curve. L* may also be chosen as a sigmoid function of the input.[8,9] It depends on the preference of the tone reproduction.

With the above definitions, we have a set of correspondence between each input RGB triplet and it's L*a*b* goal. The purpose of the neutral axis calibration is to adjust the printer to meet these goals.

Color Target Design

A color target is designed that includes 200~300 neutral and near neutral color patches. Figure 4 shows the near neutral area we are interested in. The testing colors are distributed uniformly in this area.

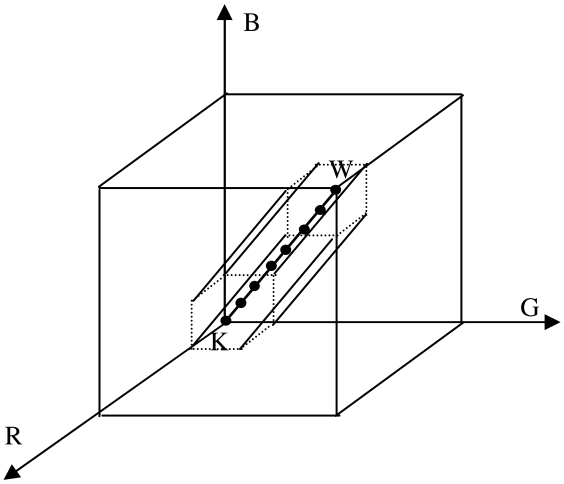

Figure 4. Neutral and Near Neutral colors

As described by Figure 4, the color patches are specified in RGB color space. We design the target in such a way that each neutral node has sufficient surrounding colors that can ensure an accurate interpolation. The CIELAB measurements of these surrounding colors are employed to calibrate the deviation of the neutral nodes.

Neutral Axis Calibration Diagram

Figure 5 shows a diagram of the neutral axis calibration. The top row is a typical pipeline of a color printer. A 3D Look Up Table (LUT), also called 3D Color Map, is utilized to make the conversion from RGB to CMYK. In ICC pipeline, you can consider it as a linked input and output profile. Four 1D LUTs control the tone reproduction curves of CMYK inks. The CMYK image is then half toned and sent to the printer.

The pre-designed color target is printed using the current 3D Color Map at the time of color calibration. The CIELAB values of each color patch are then measured, and the measuring data are stored in an auxiliary table, called 3D Color Calibration Table. Its input color space is RGB and its output color space is CIELAB. It includes only the colors defined by the test target. This table will not be used to calculate CIELAB values from RGB; it will be used to calculate RGB values from the predefined CIELAB goal. That's why we call it inverse interpolation.

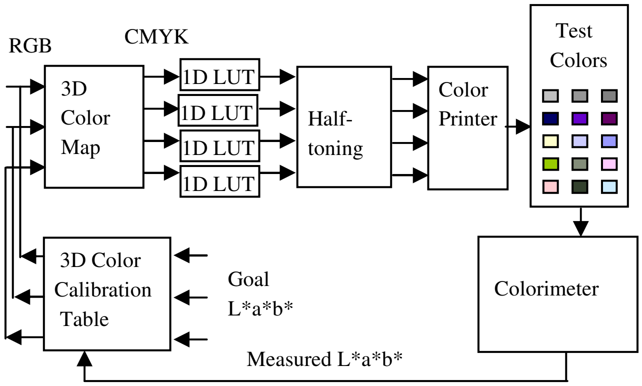

Figure 5. Neutral axis calibration scheme

The purpose of the neutral axis calibration is to calibrate the 3D Color Map in such a way that for a given RGB input, the rendered L*a*b* should hit the predefine goal. Since the 3D Color Calibration Table carries the data measured on real-time, we will use it to calibrate the 3D Color Map, which is implemented by a two-step 3D interpolation algorithm.

Two-Step 3D Interpolation Algorithm

3D interpolation is a commonly used method to perform color space conversion.[10-12] The size of a 3D LUT may be 9x9x9, 17x17x17 or 33x33x33 representing respectively 3-bits, 4-bits or 5-bits data in input color space. In order to interpret 8-bits input data, a 3D interpolation is required that converts the image from an input color space (e.g. RGB) to an output color space (e.g. CMYK).

Two 3D LUTs are involved in the neutral axis calibration procedure. One is the original 3D Color Map, which is the color map to be calibrated. The other is the 3D Color Calibration Table, which is an auxiliary table used to store the measuring data. As described above, the input of the 3D Color Calibration Table is RGB, and the output of this table is CIELAB.

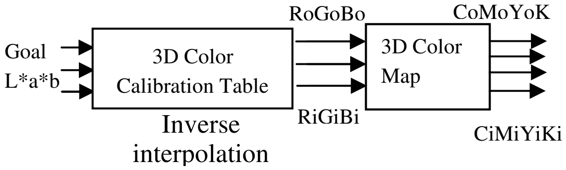

Figure 6. Two-Step 3D Interpolation Diagram

The whole calibration procedure consists of an inverse interpolation of the 3D Color Calibration Table, cascaded by a forward interpolation of the 3D Color Map. Figure 6 illustrates a diagram of this procedure.

For a given neutral node RiGiBi, we take its goal L*a*b* as the input, and use the 3D Color Calibration table to inversely interpolate the corresponding RoGoBo. If RoGoBo is not equal to RiGiBi, this node is deviated from the pre-defined neutral axis. Then we use RoGoBo as the input and to perform a forward 3D interpolate using the 3D Color Map, and obtain CoMoYoKo as the output. After these two interpolations, we know that if we use CoMoYoKo inks to print RiGiBi, the printing result will hit the goal L*a*b*. Therefore we replace the CiMiYiKi values in the node RiGiBi by CoMoYoKo in the 3D Color Map. In other words, before the calibration, an input RiGiBi was converted to CiMiYiKi, and the printing result was deviated from the goal L*a*b*. After the calibration, the input RiGiBi was converted to CoMoYoKo, and the printing result will exactly hit the goal L*a*b*.

There are several different methods to do 3D inverse interpolation: tri-linear, pyramid, prism and tetrahedral [10]. The main shortcomings of these methods are: the interpolation error is pretty high and under certain conditions the equations are insolvable. We prefer to use a forward exhaustive search method to perform inverse interpolation. In a given cube, we compute the L*a*b* values for all possible RGB grids with a sufficiently small step. Once we find an L*a*b* value which has the minimum ΔE with the goal L*a*b*, its corresponding RGB is the result of the inverse interpolation. Because we have only a limited number of nodes to calculate, this method can provide very robust and very accurate result in an acceptable time.

This procedure is performed for every node in the neutral axis except the black point and the white point. Because the black point is often constraint by the ink limit, we can't modify it freely. The white point represents the paper white; we have no means to change it. In a 3D LUT of the size NxNxN, only N-2 neutral nodes are calibrated.

It is to note that if we calibrate only the neutral nodes, it may produce some discontinuity in the color map especially for the near neutral colors. A smoothing algorithm is used to solve this problem. For each near neutral node, we select a small number of neighboring nodes including one or several calibrated neutral node(s), and use a weighting average function to smooth this node.

Experimental Results

Figure 7 shows a neutral ramp that was used to test the neutral axis calibration algorithm. Before the calibration, the 3D Color Map was manually adjusted, the gray colors were bluish. After the calibration, the ramp becomes more neutral and more consistent from light to dark.

Figure 8 shows the L*a*b* values measured for the neutral ramp before and after calibration, where we show respectively the calibration goal, the values before and after calibration. Comparing to the goal L*a*b*, it can be seen that before the neutral axis calibration, the color differences were up to 6~7 △a, △b, and after the calibration, the color differences are reduced to 1~2 △a, △b. The calibration errors are mainly due to the measurement precision and the rounding errors of the 3D interpolation.

Figure 7. Neutral ramp with 17 patches.

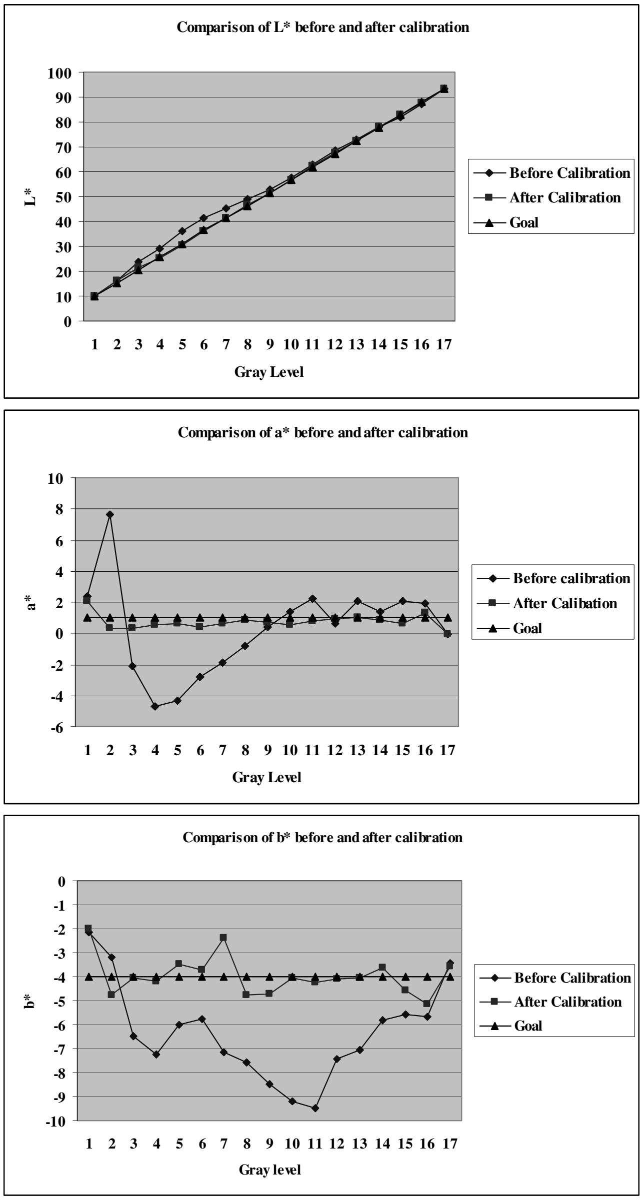

*Figure 8. L*a*b* values before and after the neutral axis calibration.*

The computational time of the neutral axis calibration algorithm is about 2 seconds for a Pentium II 450MHz PC. This time is negligible if we compare it with the total calibration time. Most of the color calibration time is spent for the color measurement. It may take 3~10 minutes to measure the color target depending on what measuring instrument is used.

Conclusion

A neutral axis calibration method is proposed. The basic idea is to define an L*a*b* goal for each RGB node in the neutral axis; print and measure some neutral and near neutral color patches; and then run a 3D inverse interpolation algorithm to calibrate the printer color map. It provides an automatic and accurate way to calibrate the neutral axis of a color printer. The advantage of using a common L*a*b* goal is that we can ensure not only the color consistency for a same printer at different time, but also the color consistency over different printers.

We have presented a very robust and accurate 3D inverse interpolation algorithm which is particularly important for online color calibration.

This method is not limited to the calibration of the neutral colors. For any other colors such as skin tone colors or other memory colors, if you can define a goal L*a*b* for a given input RGB, you can always use this method to perform the color calibration.

References

1. CIE Technical Report, Industrial Colour-Difference Evaluation, CIE 116-1995, pp. 9, ISBN 3 900 734 60 7, 1995.
2. W. Wu and R. Rasmussen, "Color calibration techniques for print quality measurements", Proc. IS&T's 2001 PICS Conference, pp. 76-79, 2001.
3. S. Livens, M. Mahy and D. Vansteenkiste, "How to ensure consistent colour quality in inkjet proofing", Proc. SPIE, Vol. 4663, pp.130-136. 2002.
4. T. N. Lin and J. Shu, "A Color consistency algorithm between different printers", IS&T's 1998 PICS Conference, pp. 409-411, Portland, OR, 1998.
5. T. Balasubramanian, M. S. Maltz, "Method for refining an existing printer calibration using a small number of measurements", US Patent 5739927, April 14, 1998.
6. Y. Wu, "Methods and arrangements for calibrating a color printing device using multi-dimensional look-up tables", US Patent Pending, Publication No. 20020180998, December 5, 2002.
7. R. Bala, "What is the chrominance of 'Gray'? ", IS&T and SID Ninth Color Imaging Conferences, pp. 102-107, Scottsdale, 2001.
8. G. J. Braun and M. D. Fairchild, "Image lightness rescaling using sigmoidal contrast enhancement functions," *Color Imaging: Device Independent Color, Color Hardcopy, and Graphic Arts IV, Proc. SPIE 3648,* 96-107, 1999.
9. C. H. Wen, J. J. Lee, and Y. C. Liao, "Adaptive quartile sigmoid function operator for color image contrast enhancement" , The 9th Color Imaging Conference, pp. 280-285, Scottsdale, Arizona; November 6, 2001.
10. H. Kang, "Color Technology for Electronic Imaging Devices", SPIE Optical Engineering Press, ISBN 0-8194-2108-1, 1996.
11. K. D. Gennetten, "RGB to CMYK conversion using 3-D barycentric interpolation", Proc. SPIE, Vol 1909, pp116-126, 1993.
12. J. Kasson, W. Plouffe, S. Nin, "A tetrahedral interpolation technique for color space conversion", Proc. SPIE, Vol. 1909, pp127-138, 1993.

Biography

Yifeng Wu received his BS degree from Tsinghua University in China, in 1982. He received his MS and Ph.D degrees from École Nationale Supérieure des Télécommunications (ENST) in Paris, France, in 1984 and 1987 respectively. He was a research associate at ENST from 1987 to 1991, where he developed the color acquisition and rendering system for VASARI project. From 1991 to 1999, he worked as a software engineer and color scientist at Océ, France. He joined Hewlett Packard in 1999. His main interests focus on color science and image processing.