# Validating the black point compensation standardization

*Marti Maria Saguer; Hewlett Packard Española; Sant Cugat del Vallés, Catalonia, Spain*

## Abstract

*Black point compensation is a widely used feature when using the relative colorimetric intent to transform images. This procedure was first implemented in Adobe Photoshop® in the late 1990's. This implementation is described in "Adobe Systems' Implementation of Black Point Compensation" dated 2006 and available on the Adobe website. The International Color Consortium (ICC) has recently created an updated description of this algorithm to allow black point compensation to be used in a consistent manner across applications and to provide a close match to results obtained in Photoshop with the Adobe color management module (CMM). The new document includes corrections that weren't addressed in the original Adobe paper. A number of tests have been conducted in order to check the suitability and conformance of the revised algorithm and description. In this paper, a summary of the test implementation and the checking done so far will be presented.*

## Introduction

Black point compensation (BPC) is a technique used to address color conversion problems caused by differences between the darkest levels of black achievable on different media/devices. Although ICC profiles specify how to convert the lightest level of white from the source device to the destination device, the ICC profiles do not specify how black should be converted. The purpose of BPC is to adjust a color transform between source and destination ICC profiles, so that it retains shadow details and utilizes available black levels of the destination device.

Because BPC is an optional feature that the user can enable or disable when converting an image, the user can always decide whether the conversion of a particular image looks better with or without BPC. This makes the entire process a question of preference and therefore a perceptual issue.

BPC was first introduced by Adobe in Adobe Photoshop® in the 1990's. Permission has been given by Adobe Systems Incorporated to the International Color Consortium (ICC) and ISO Technical Committee 130 (Graphic technology) to create a Technical Specification to allow black point compensation to be used in a consistent manner. The document is currently being circulated across ICC members and will soon be publicly available.

One of the main goals of the revised BPC document is to increase consistence between applications. Since there are working implementations already deployed in Adobe products, it makes sense to check how well an independent CMM would match BPC by just implementing the algorithm as described in the ICC document.

To check that, the author of this paper has implemented the black point algorithm as described by the ICC document, on top of the Little CMS[2] CMM, which is an open-source color management engine available under MIT license[3]. No previous knowledge of Adobe code has been used.

A test bed with a number of ICC profiles has been designed and executed to assess how well this independent implementation would match the Adobe CMM when performing BPC. In particular, Photoshop CS6 has been used as the reference Adobe application. Since all Adobe products share the same color engine, it is expected that the same results would be obtained by using other Adobe products.

## Background and applicability

The ICC framework[4] proposes the use of profiles associated with devices and/or content. It provides the ability to communicate color via a Profile Connection Space (PCS), representing colorimetry (e.g. CIE XYZ or L*a*b*), the *lingua franca* among all proprietary device representations of color. Thus an image's color is interpreted thanks to an associated source profile and employing a color management engine it can be transformed to a destination color via the intermediate PCS. A fundamental principle of this workflow is that a device's profiles are independent and agnostic of other devices and a transformation between any two is defined. The key to this mechanism is thus the intermediate, common PCS.

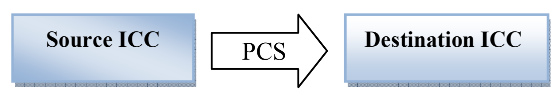

***Figure 1.** ICC Color Management communication via a common PCS.*

The BPC procedure depends only on the rendering intent(s) and the source and destination ICC profiles, not on any points in a particular image. Therefore, the color transform using specific source and destination ICC profiles can be computed once, and then efficiently be applied to many images which use the same ICC profile color transform pair.

Not all profiles and not all intents are suitable to be used with BPC. Namely, absolute colorimetric intent (either the new ICC-absolute or the old V2-absolute)[7] does not apply. Also, device link or abstract profiles cannot be used. This is due to the true nature of the BPC algorithm and device link ICC profiles. Since BPC is basically a remapping of how profiles are connected, a device link which includes already connected profiles cannot be used at all.

## The algorithm

Adobe's BPC is basically a linear scaling in the XYZ colorimetric space. At this point, it is important to note that the XYZ space is not perceptually uniform. BPC implemented as a linear scaling in the XYZ space, moves colors perceptually non-uniform across the lightness axis. This effect can be regarded as convenient since it keeps most of the gamut almost untouched and only noticeably moves the colors in the dark shadows. Other implementations have been using different alternatives[6], like sigmoidal compression on J axis using modern perceptual spaces like CIE CAM02.

Anyway, the goal of this paper is to expose the results of the qualification tests, and not to discuss the suitability of the Adobe algorithm. A plain rescaling using XYZ is the approach used by Adobe and that is the adopted method in the ICC paper as well.

The algorithm can be split in two steps. The first step consists in obtaining all needed information. The second step is to compute a modified color transform that would be used to convert the desired image(s).

When concatenating two ICC profile to build a color transform, the PCS can be either be CIE L*a*b* or CIE XYZ[4]. The CMM can force the color transform to use the XYZ space as PCS because it is convenient for BPC. Conversions from/to CIE L*a*b* to XYZ are already necessary for proper profile connection, so all capable CMM need to have this functionality. A step with a rescaling of XYZ is then inserted in the middle of PCS to implement BPC.

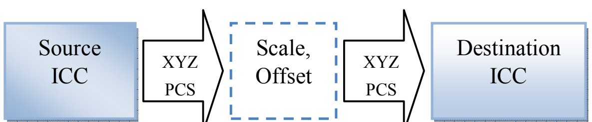


***Figure 2.*** *Rescaling done in the XYZ PCS*

The transformation is built in a way that the source white is mapped to the destination white and the source black is mapped to destination black. Colors are linearly mapped by:

$$XYZ_{DST} = scaleXYZ * XYZ_{SRC} + offsetXYZ$$

As we want to build a transform that adjusts the dynamic range, we need endpoints for both source and destination media. Those are the maximum and the minimum values source and destination may have, and correspond to media white and black points. ICC profiles uses relative colorimetry and, as said, absolute intents are not supported in combination with BPC, so white points of both source and destination are always assumed to be D50. To calculate *scaleXYZ, offsetXYZ*, we just need to solve the following equations:

$$XYZwhiteD50 = scaleXYZ * XYZwhiteD50 + offsetXYZ$$
$$XYZblackDST = scaleXYZ * XYZblackSRC + offsetXYZ$$

Unfortunately, obtaining the black points *XYZblackDST* and *XYZblackSRC* is not so easy: it turns to be fairly complex due to several factors, which include buggy profiles, poorly defined specs and deprecated tags.

## Black point detection

As discussed previously, the most complicated part of the Adobe algorithm is to detect the ICC profiles black point. Version 2 (V2) of ICC spec[5] defined some time ago a tag holding the measured media black. This tag was optional, so there was no guarantee that a given profile would have it. And unfortunately, on a survey conducted by the ICC the tag was found to be buggy and unreliable in many ICC profiles, so all CMM were ignoring it. Because of that, the ICC deleted the entry in the version 4 (V4)[4] specification, so black point as a tag is no longer supported.

Instead, CMMs are supposed to detect the black point of each profile and each rendering intent by their own methods. The ICC BPC document discloses a number of ways to perform this task. This is useful far beyond BPC. Complexity of those sub-algorithms varies from the simplest one which is just to convert darkest colorant to CIE L*a*b* by using the profile, to parabolic curve fitting required for noisy output profiles.

For perceptual and saturation intents, we have to differentiate between V2 and V4 ICC profiles. V4 profiles and perceptual/saturation intents are actually the easiest case. Since ICC specified a fixed value perceptual black, we just need to return this value when a proper combination is detected. For V2 ICC profiles the process is however far more complex. In well-behaved V2 profiles, perceptual and saturation black points turns to be CIE L*a*b* (0, 0, 0). However, the V2 specification was not so clear about when to rescale dynamic range in perceptual intent, as a result there are a number of V2 profiles that uses black points different from zero in perceptual or saturation intents. Detection of those black points is performed in a similar way that the relative colorimetric intent.

The next way to detect black point is to just use the profile in reverse direction, and provide the darkest possible colorant. For example, for a RGB ICC profile using CIE L*a*b* as PCS, we could evaluate the value RGB (0, 0, 0) across the AToB1 tag to obtain the L*a*b* value associated with RGB black. This works to some extent, assuming the profile is well behaved; it has no noise and is suitable for input. Display and RGB/Gray color space profiles can use this method.

For CMYK this is no longer valid since CMYK devices are usually ink-limited. For CMYK and multi-ink spaces, a round-trip L*a*b* → Colorant → L*a*b* must be used. The first conversion L*a*b* → Colorant computes the colorant associated to L*a*b* value of (0, 0, 0) by the ***perceptual intent***. This returns the darkest ink-limited colorant combination as know by the profile. The next step is to get the real L*a*b* of this colorant, and this can be obtained by the Colorant →L*a*b* conversion by using the ***relative colorimetric intent***, which corresponds to the BToA1 tag. This effectively takes care of any ink-limit embedded in the profile. CMYK profiles used as input can use this method.

When an ICC profile based on 3D CLUTs is used as output, no matter whether RGB or CMYK, some additional processing is required. In the case of output profiles, the output direction is what the profile is really designed for, and usually it holds a resolution much higher than the input direction. The input tables AToBxx are often simplifications of reversed BToAxx, since the goal of those tags is mostly to provide soft-proofing capabilities. It is certainly possible for such output profiles to have small differences in the output direction, and even, to have "noise" near to the dark shadows. This effect is seldom seen in the proof direction.

For those profiles, the Adobe algorithm requires to fit a least squares error quadratic curve, as seen in figure 3. Note: Marti you could explain how you pick the darkest point according to Figure 3. The associated math is not too complex, but above the scope of this paper. Details can be found in [1]

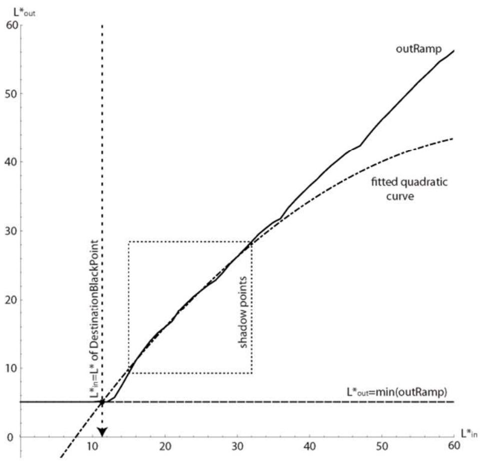


**Figure 3.** Example curve fitting

## The test bed

Since the main goal of this standardization is to increase consistency, one of the first things to check is whatever the published algorithm is consistent with Adobe products. By using an independent CMM not affiliated with Adobe, we make sure developers will have all information to successfully create a consistent CMM. To check the feasibility, we implemented the algorithm as described by the ICC document and then designed and executed an extensive test bed. The tests were aimedat two different goals:

- To check robustness of the algorithm.
- To check consistency with the Adobe color engine.

The selected CMM was Little CMS[2], which is a well-known open source color management engine. Little CMS is distributed under the MIT[3] open source license. This makes it especially suitable to build prototypes that may end in commercial products. A comparison with the Adobe color engine was performed by using Photoshop CS6 as host application.

A program coded in the "C" language and Photoshop scripts were used to automate the process. Checks included transform creation from a known profile RGB profile (**sRGB IEC61966-2.1**) and CMYK (**U.S. Web Coated SWOP v2**) to every single profile in the test. We used the profiles in the output direction to check all ways of black point detection. Each profile was used in relative colorimetric, perceptual and saturation intents.

308 assorted ICC profiles were used to create the test. Some of those profiles are collected from known vendors. Others are of unknown origin and were collected from the internet. The distribution of the test bed according device class and color space is shown in tables 1 and 2. Some of those profiles belong to classes "abstract" or "named color". For those classes, the algorithm was expected to refuse to perform the BPC operation. No broken profiles were used, although some of the profiles in the list were slightly non-compliant in the sense they missed some tags.

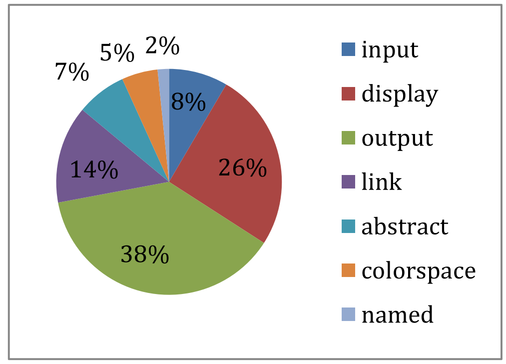


***Table 1:*** *Classification of sample profiles according class*

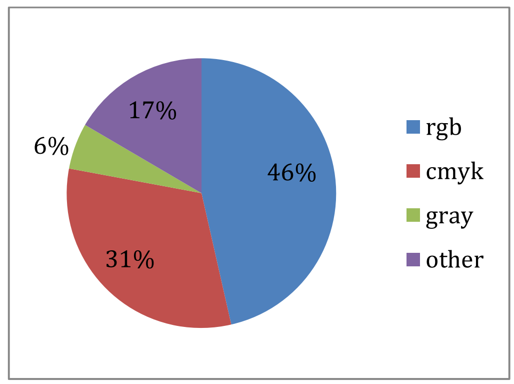


***Table 2:*** *Classification of sample profiles according color space*

For assessing image quality, a photographic image (*Figure 4, right*) and a drawing (*Figure 4, left*) were used. Bitmaps obtained after re-rendering from Photoshop and Little CMS were compared pixel by pixel and the maximum, average, 95% percentile and standard deviation were reported.

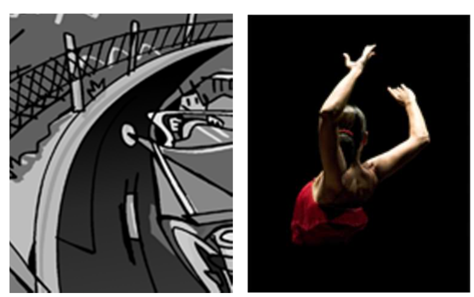

***Figure 4.*** *The test images used to evaluate smoothness of BPC algorithm.*

The profile classes suitable to be tested are input, display, output and color space. This makes a total of 238 profiles, which means 714 single tests if we check all 3 intents for each profile.

## Results

The algorithm implemented on top of the Little CMS framework ran seamless on all profiles, discarding unsupported ones and detecting the black point by using different methods. No major issues were found, despite the test uncovered a minor bug in the code. This turned to be an error in the implementation instead of an issue of the ICC document. After fixing the code, the rest of test executed ok.

In all cases, differences were under 3 digital counts per channel, which can be explained as different round strategies of both CMM. Sample plots showing distribution of differences in test images can be seen in *figure 5*. Black dots are differences of 1 digital count in the K channel. No further differences were found in this case. In many other cases there were no differences at all. The test was using SWOP as destination profile and sRGB as source.

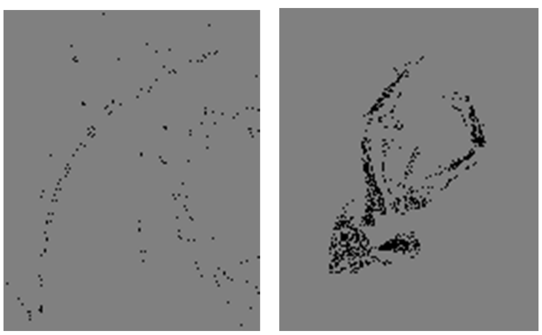

***Figure 5.*** *Differences between Photoshop CS6 and Little CMS on the black channel*

Despite the collection of sample ICC profiles is not aimed to represent any special usage or source, is interesting to note the big number of profiles that were found "noisy" and therefore needed a curve fitting method to detect the black point. For 714 test cases (which can be up to 3 per ICC profile, one for each intent), 186 were found "noisy", which corresponds to roughly 26%

This is a very high proportion for a black point detection method that seems mainly aimed for buggy profiles. Whether this conclusion can be extrapolated to all profiles is out of the scope of this paper, but the fact is such method for black detection on noisy ICC profiles is needed and used in many real-world cases.

## Conclusions

Black point compensation is a technique used to address color conversion problems caused by differences between the darkest level of black achievable on one device/media and the darkest level of black achievable on another. The International Color Consortium (ICC) and the ISO Technical Committee 130 (Graphic technology) have created a document describing an algorithm to allow black point compensation to be used in a consistent manner across applications. A number of qualification tests have been performed using this algorithm. The tests have found the results to be robust and highly consistent with the black point compensation feature offered by Adobe products.


## Acknowledgements

The author would like to thank Lars Borg and Chris Cox from Adobe, as well as Dave McDowell and the whole ICC for their support.


## References


[1] Adobe Systems Incorporated. (2006) Adobe Systems Implementation of Black Point Compensation. Available from: http://www.color.org/AdobeBPC.pdf

[2] M. Maria. Little CMS Engine – How to use the Engine in Your Application, 2012, Available from http://www.littlecms.com

[3] MIT license (also called X11 license) available at http://www.opensource.org/licens es/mit-license.php

[4] Specification, I. C. C. (2004). 1: 2004-10 (Profile version 4.2. 0.0). International Color Consortium.

[5] Specification, I. C. C. "1: 1998-09." File Format for Color Profiles,''International Color Consortium (1998).

[6] Yerin Chang -"Evaluation of preferred lightness rescaling methods for colour reproduction", Proc. SPIE 8293, Image Quality and System Performance IX, 82930G (January 24, 2012)

[7] Kriss, Michael. Color management: understanding and using ICC profiles. Ed. Phil Green. Vol. 17. Wiley, 2010.


## Author Biography


*Marti Maria is a color engineer at the large format printer division of Hewlett-Packard. Marti is also the author of well-known open source color oriented packages, like the Little CMS open CMM and the LPROF profiler construction set. He has contributed to several color books and was session chair at the 16th IS&T/SID Color Imaging Conference.*