

# Natural Color Profile Adjustment for Professionals

Peter Morovič, Ján Morovič, Jordi Vilar, Jordi Arnabat, Ángel Albarrán  
Hewlett-Packard, Sant Cugat del Vallés, Catalunya, Spain

## Abstract

*While there may be no point in arguing about taste, creative professionals make a living from sharing theirs. Making specific, individual color preferences that a creative professional knows how to achieve when creating content on a display also propagate into print is a significant challenge since it lacks real-time feedback. The present paper introduces a method for allowing creative professionals to use the tools they know and love to also personalize the color behavior of their devices. This is achieved by analyzing color changes applied to images and applying them to a device's ICC profile. As a result the personalized device results in customized color behavior regardless of the workflow used. The paper describes the ICC profile transformation algorithm in detail and provides a color error analysis of its performance.*

## Introduction

Color reproduction is not a 1:1 task. For a given original color or image there are typically a myriad alternative reproductions, each favoring some aspect of the original over another or being preferred by one person versus another. This is a well-known attribute of color reproduction that has led to the definition of a variety of color reproduction objectives, such as those of Hunt [1], Yule [2] and others.

Not only can different types of reproduction objectives be followed, but individual preferences also come into play especially when color reproduction is done by creative professionals like photographers or artists. In fact the specific trade-offs made by a creative are part of their signature look and are integral to the creative process. In addition to such individuality of preference, the language used to express it varies from one person to another, as do the corresponding expectations. An artist may instruct their print maker to make print look 'more creamy,' which is a peculiar way of specifying appearance to begin with and which translates to specific appearances as a function of the artist (one artist's 'creamy' may be another's 'silky').

Such richness and variety of language may suggest flexibility of interpretation, but the opposite is in fact the case. A specific look is sought and practically unlimited time is available to achieve it. Perfection is the name of the game. Fortunately, creative professionals have the skills to make their desired look happen using tools like Adobe Photoshop or Apple Aperture.

The challenge then comes when color adjustment is complete on a display, where real-time feedback allows for precise, iterative fine-tuning and the first print is made. At this point a color change is introduced that is constrained by color gamut differences, but where single, fixed trade-off (albeit defined by engineers in collaboration with artists) is applied. Any changes to this trade-off need to be done without the real-time feedback, are substrate and printer specific and result in an awkward user experience.

To address the above scenario, this paper presents an approach to printer or scanner color personalization that allows for the use of arbitrary image color editing tools (e.g., the ones a particular professional favors) and via a transformation detailed

below applies their color changes to a device's ICC profile, thus making the changes permanent for the device and applied transparently from the user's point of view.

## State of the Art

Personalizing the color rendering of a device is nothing new, and there are numerous solutions that provide specific tools for it. Most of them allow for the editing of ICC profile tags directly (e.g., tone curves can be adjusted, LUT entries tuned by hand) or provide sliders labeled 'contrast', 'saturation' and 'hue' that allow for global color changes to be applied. Lastly, the profile output can also be edited in the output space, e.g., via CMYK tone curves. While these tools can be very effective, their target are not creative professionals, but instead pre-press operators. To a fine artist CMYK sliders are about as accessible as the distinction between Art Nouveau and Jugendstil is to the average press operator.

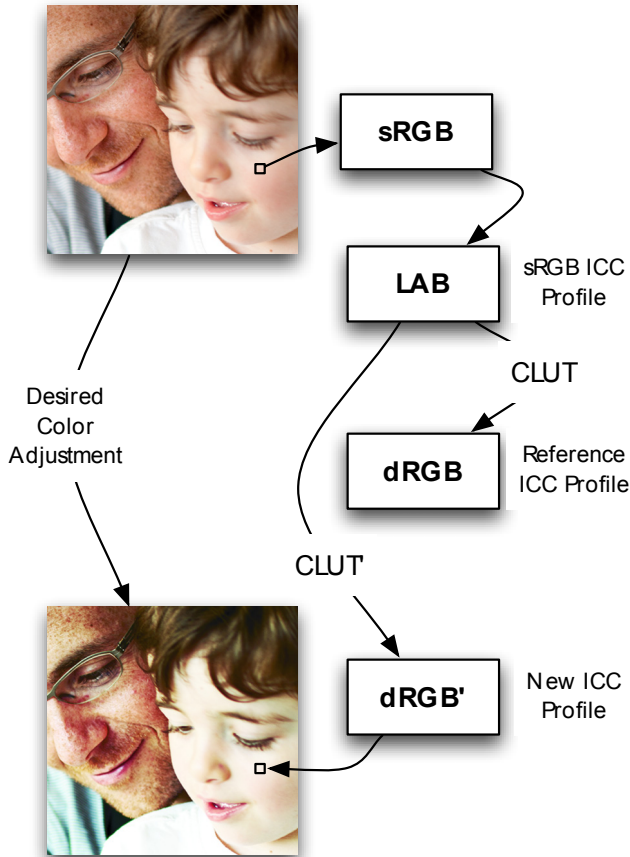
It is also worth noting that the 'natural language' approach of Woolfe [3] is complementary to the work presented here. The basis of their solution is that non-experts use everyday language (a tautology) and that color changes expressed in it can have specific color transformations assigned to them. The key to this working is that non-experts not only use 'common' language, but also that their expectations are categorical in some sense. That is, they don't expect a specific result, but instead will accept a result from the right category of possible corresponding transformations. In other words, lax but shared specification is coupled with broad acceptability so that a request to "make the sky more saturated" can be acted upon by some fixed adjustment of saturation applied to a range of blues. Professionals on the other hand tend to have lax and individual ways of expression what they want, but very specific expectations and a narrow range of what they will accept, so the earlier example of a more "silky print" may translate to very different sets of adjustments that range in nature, magnitude and complexity from artist to artist.

## Encapsulating color adjustments in a profile

The basic idea of our approach is to take a user's image before they make a first print and then let them make changes to their image that they would like to see in the print (i.e., edit the printer's color behavior by showing how it ought to change on their image). Once the image is successfully adjusted to yield desired print colors, a transformation is computed from the before and after images and applied to the 'default' output ICC profile. Printing with the user's color preferences can then be had from any application that supports ICC color management (including printing via a RIP) and not only from the one where the color changes were specified. In more detail the process is as follows:

To adjust the color rendering of a printer in a visual way, an image is chosen first. This can be an image that has a particular color palette for which the default ICC profile output is not to the liking of a user, or a photograph that represents the conditions of a particular occasion (e.g. the lighting, the intended 'mood'), or even an artificial chart-like image to change some or

all colors in a visual way. The image is then opened in an image editing software (e.g., Adobe Photoshop, Apple Aperture, etc.), and the user adjusts the colors to their liking. Ideally this would be done in soft-proofing mode (i.e., previewing on a monitor what the image content would look like on the device whose profile is used), thus giving the user a more accurate visual feedback of what to expect. A key point here is that any color image editing software can be used and that within each of these applications the user can adjust colors however they like. From studying professionals it can be seen that they each tend to have different methods for getting the same result (e.g., correcting for a color tint by means of the Hue/Saturation interface of the Color Balance one in Photoshop, which can be used to identical effect). Allowing them to use the tools they are already familiar with lets them get to the desired outcome more quickly and with more confidence than if a new interface (e.g., with sliders) were used.



**Figure 1:** The desired color adjustment from the top image to the bottom image is encoded as an ICC profile modification on top of a reference ICC profile, by means of modifying the BToA tag's Color Look Up Table.

Let us then denote  $I^u$  as the unadjusted image and  $I^a$  the adjusted image and assume that they are both in a native color space associated with them (i.e. they are tagged with an ICC profile, such as sRGB/Adobe RGB). This is a safe assumption since creative professionals are strong adopters of color-managed workflows. In the absence of such tagged profiles, a default (sRGB) can be used. Let us then consider an output profile  $P^r$ . The objective of this approach is to create a new ICC profile  $P^N$ , based on the chosen reference ICC profile  $P^r$  such that given a Color Management Module (CMM) that can apply profiles to images:

$$CMM(P^r, I^a) = CMM(P^N, I^u) \quad (1)$$

In other words, an adjusted image  $I^a$  with the reference profile  $P^r$  applied (or soft-proofed) using a CMM should match the unadjusted image  $I^u$  with a newly computed profile  $P^N$  that represents and embeds the relationship between the unadjusted image  $I^u$  and the adjusted image  $I^a$ .

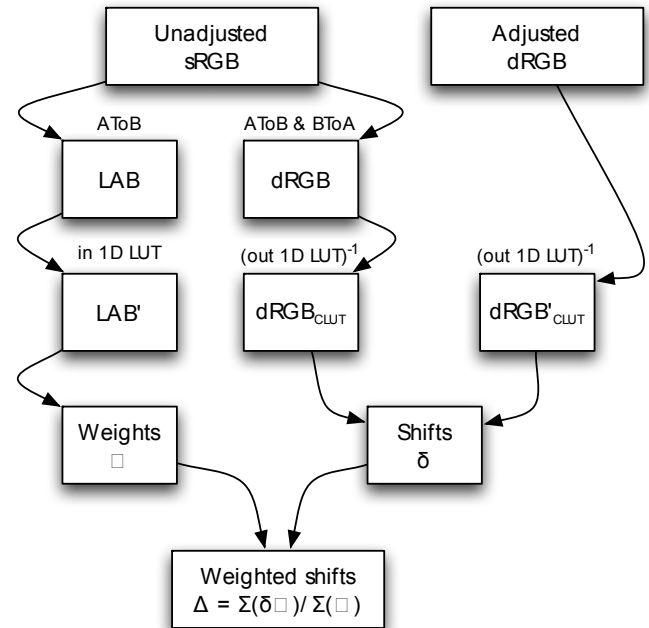
A perfect match will not always be possible due to both the nature of ICC profile color transformation mechanisms, which have limitations, and the high level of control afforded by image editing software. However, as shown in the results section below, a very good degree of correspondence can be achieved in practice.

A key realization here is that ICC tagged image content gives direct access to colorimetry via the Profile Connection Space, which furthermore is indexed using a regular cubic mesh encoded in the Color Look-Up Table (CLUT) of an ICC profile making it fast to locate nodes involved in the interpolation of a given pixel. Hence, differences between unadjusted and adjusted images can be included back into the CLUT taking into account how the CLUT will be used to interpolate each color pixel, to recover the desired adjustment shift.

This naturally results in the following benefits:

- only the image gamut is taken into account;
- the algorithm is single-pass (linear cost with image size);
- the changes take into account the interpolation of the CMM;
- the changes are weighted in the perceptually near-uniform CIELAB space.

A schematic overview of the algorithm is shown in Fig. 2 below, with a detailed description thereafter.



**Figure 2:** A schematic overview of the algorithm that computes weighted shifts from a pair of unadjusted and adjusted images. The weighted shifts are then applied to the reference ICC profile CLUT in order to create a new CLUT.

1. Convert each original (unadjusted) image pixel to the PCS (without loss of generality let us assume this is CIE LAB) using the reference profile  $P^r$  and then apply the input curves of the BToA table(s) of  $P^r$ .

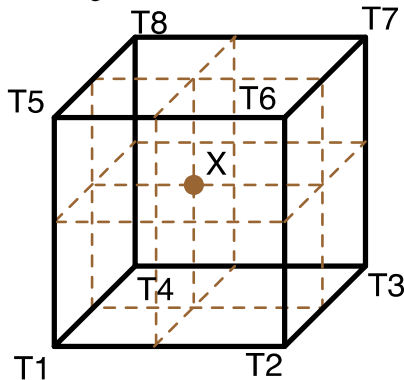
2. Compute the indices of the 8 neighboring nodes (those involved in the interpolation of a pixel's PCS value, assuming tri-linear interpolation – see Figure 3), and the weights  $\sqrt{\phantom{x}}$  used for interpolation.
3. For each pixel in the unadjusted image, convert it to  $P^r$ 's device space.
4. Apply the inverses of the output curves of the BToA table(s) to all channels of the reference device space.
5. For each pixel in the adjusted image (already in the reference device space), apply the inverse of the output curves of the BToA table(s) for all channels of the reference device space.
6. For each pixel and channel obtained from steps 3 and 5, compute their difference vector  $\Delta$ .
7. Compute a weighted shift  $\Delta$  for every channel of each node of the CLUT, and apply it to the CLUT in order to obtain the adjusted profile.

The final result is a weighted shift applied to all nodes of  $P^r$ 's BToA CLUTs that have been affected by the user's adjustments. Once the process is finished, the resulting profile will contain an adjusted response optimized for the reference profile.

The LAB values of the unadjusted image,  $I_{LAB}^u$  directly define indices into the cube tessellation of the CLUT. Suppose a CLUT has  $N$  nodes and encodes 16 bit values, the LAB image is converted at each pixel into a CLUT index image by dividing each LAB by the sampling step of the CLUT ( $s$  = maximum value at 16 bit encoding divided by  $N-1$ ) as follows:

$$\tilde{I} = I_{LAB}^u / s \quad (2)$$

Then, for each pixel, the index image – a floating-point value on the range of  $[1, N]$  – directly defines the enclosing cube in the CLUT. This is computed by means of finding the floor and the ceiling of the index values, with care taken at the edges of the CLUT. Hence for each pixel we obtain a set of 8 coordinates, corresponding to the vertices of the enclosing LAB cube, as shown in Figure 3 below.



**Figure 3.** The enclosing cube of an LAB value  $X$ , with vertices  $T1$  to  $T8$ , and the projections of  $X$  onto the faces of the cube - these used in tri-linear interpolation to obtain the RGB output value corresponding to  $X$ .

Vertices  $T1$  to  $T8$  are the ones used for interpolating the LAB value  $X$  that gave rise to the RGB in the unadjusted image and consequently, these are the nodes we want to encode the change to the adjusted color of  $RGB'$ . Since there can be multiple pixels in an image that fall within the same cube, we have a set of points  $X_r$  with corresponding  $RGB_i \rightarrow RGB'_i$  shifts and we want to take all shifts into account. Once the whole image is parsed and the RGB shifts as well as their corresponding weights are computed, a weighted average is

stored at each node of the CLUT. The shifts are simply difference vectors in the CLUT RGB space:

$$= [RGB_i - RGB'_i] \quad (3)$$

While the weights  $\sqrt{\phantom{x}}$  are the relative weights of the vertices  $T1$  to  $T8$  that would have been used to interpolate the location of the LAB value corresponding to the RGB – these are relative to the location of the cube vertices and are computed in LAB space, hence giving perceptual relevance to the distances.

After computing the weighted shifts of the whole CLUT, all nodes involved in interpolating any of the image pixels have been adjusted. While this is fine for the reference image itself, in order to make sure the profile is applicable to other images with the same adjustments and robust to variation in image content, it is important to ensure no discontinuities have been introduced. To do this, a smoothing of the CLUT nodes is applied after all the shifts computed from the unadjusted vs. adjusted image are applied. Note that a desirable constraint is to maintain the effect of the computed ICC profile on the reference image, so any smoothing should not affect the CLUT nodes which have been adjusted or are contained in the image and have been left unadjusted (no adjustment of image gamut colors means the user is happy with the colors of that region). Instead a smoothing should affect surrounding, neighboring nodes only. So, for example, if the unadjusted image has two close but disconnected blue regions that a user has effected similar shifts upon, it is not desirable to preserve the intermediate nodes as unchanged even if no change was effected in these (as a consequence of the image gamut variety) since another image that does have these colors represented would be expected to have those nodes consistently adjusted as well. The benefits of applying this post-processing in the form of a CLUT node smoothing are the robustness of the newly computed CLUT when applied to other image content.

A brief description of the smoothing approach, applied after the CLUT adjustment has been computed in the earlier described fashion, follows:

1. For all nodes of the CLUT check if the node has been adjusted or not (we also consider as 'adjusted' the nodes where the adjustment results in a shift of 0 – i.e. image content that the user didn't modify).
2. If the node has been adjusted, move on to the next node.
3. If the node has not been adjusted, let's denote it as node  $X$ , find all  $N$  neighbors of node  $X$  (in a cube mesh a single node has 26 neighbors)
4. For each of the  $N$  neighboring nodes to  $X$ , check if they have been adjusted or not
5. If a node is found to have been adjusted, move on to the next of the  $N$  neighbors
6. If the neighboring node has not been adjusted, propagate the adjustment found in  $X$  that is proportional to it's distance to  $X$  (e.g. diagonal nodes are further away then orthogonal neighbors) to the current adjusted node (from step 3). In weighing the propagated adjustment, both the distance (in CIE LAB) from node  $X$  and the number of un-adjusted neighbors is taken into account.

The nodes, which have been adjusted are not modified (smoothed) since, if an adjustment was made by a user for a set of nodes in a certain area of color space (e.g. if the blues of the sky have been adjusted in a certain way), what remains to be

resolved are the frontiers of this region, the areas of blue that are not sky-blue but use nodes that neighbor with the ones adjusted.

Also, the weighing and proportional propagation of the adjustment occur, as in the CLUT adjustment algorithm earlier, in a perceptual space (CIE LAB), hence the proportion of change applied to nodes that have not been modified is perceptually proportional to the distance and/or number of nodes to the node in whose neighborhood it lies in. This is important to make sure that distances across color space are treated equally, not to introduce further unwanted artifacts.

The approach outlined above, including the CLUT adjustment computation based on a pair of unadjusted and adjusted images and the post-processing in the form of a CLUT node smoothing (without affecting the adjusted CLUT nodes), is suitable for a host of adjustment types of a color nature. These can be color-local (adjusting a subset of CLUT nodes), e.g. increasing saturation of the reds of a rose in an image, or global, e.g. reducing overall contrast. Spatial adjustments, instead, are not always possible to encode. If the spatial adjustment is consistently applied to all pixels of the same color in the image, e.g. via color-selective tools, then it is treated as if it were a color adjustment and not a spatial one (since only the image gamut is 'visible' to the algorithm). If such adjustments are spatially inconsistent, such that some pixels of a certain color are adjusted in one way, and other pixels of the same color are adjusted differently, the result of the algorithm will be an average of the two, affected both by the number of the pixels in the respective regions of adjustments as well as the magnitude of the adjustments.

## Test set-up

To test this approach, an ICC v. 2 profile was generated for a HP DesignJet Photo printer. Measurements of a 464-patch profiling target were made using the built-in spectrophotometer, measuring at 10 nm intervals between 400 to 700nm. The RGB profile for this set-up was computed using the built-in profiling engine and used as the reference profile in the two test cases below.

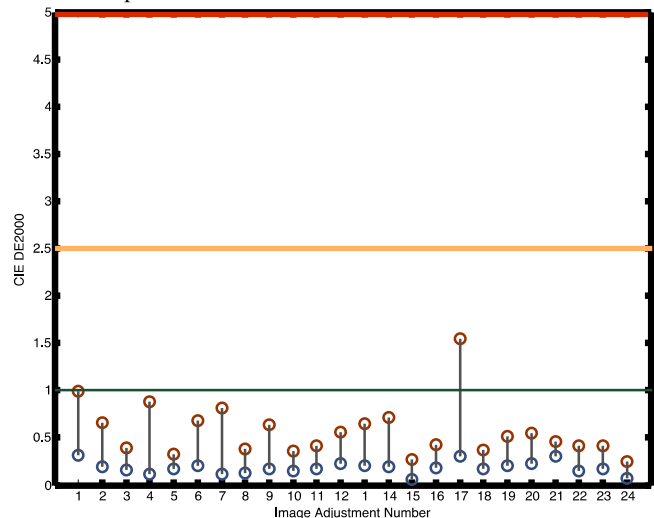
Test case 1: 24 sets of photographs from a variety of sources (both digital and analogue scans) developed by 8 professional photographers and printed using the default profiles on a HP DesignJet Z3100 on a variety of substrates, including HP Hahnemühle Smooth Fine Art (matte, photo-rag) and Photo Satin paper. For each set, the photographers were asked to make color adjustments to a sample image of their choice in order to match their preference in the printed output. The adjusted and unadjusted image pair was then used to create a new ICC profile as described earlier and its effect verified comparing the original image adjustments (in the reference profile) and the new ICC profile applied to the unadjusted image. The images were also evaluated using a color difference formula.

Test case 2: A set of portraiture and general social photography images was chosen from a single shoot, covering a variety of lighting conditions. The resulting raw captures of a Nikon D90 camera were developed to the liking of the author in Adobe Lightroom 3.0 and printed using a default profile for a HP DesignJet Z3200 Photo printer on HP Professional Satin paper. One of the images was then chosen as a sample and a color adjustment introduced. Again, based on the pair of unadjusted and adjusted images a new ICC profile was created using the algorithm described earlier. The effect of the new ICC profile was then looked at for the original sample image as well as other images from the set.

## Results

The accuracy with which the presented approach encodes user color image adjustments is examined by comparing images adjusted in an image editing application and soft-proofed using the reference profile (i.e., what a user would like to see in print) versus the unadjusted image, converted using the newly processed profile (derived from the unadjusted and adjusted image pair) – the algorithmic solution.

An alternative implementation of the algorithm is to work directly in the device color space, instead of in the source color space with soft-proofing. However, this limits the control a user has over the way the source-gamut is mapped to the destination gamut, which is part of the reason one might wish to adjust the profile. This difference has no effect on the algorithm though and simply changes the domain within which the RGB color shifts are expressed.



**Figure 4.** Color differences (CIE  $\Delta E$  2000) between adjusted images and unadjusted images converted to the modified ICC profile shown as median and 95<sup>th</sup> %tile over all image pixels.

Figure 4 shows the CIE E2000 difference terms between the adjusted images interpreted using their original ICC profiles and the unadjusted images converted using the new, processed ICC profile. The blue circles correspond to the medians and the red circles to the 95<sup>th</sup> %tiles over all image pixels in each of the respective examples. This is done for 24 adjusted images including different image content, different adjustments and different ICC profiles. For reference, the green line at 1 E 2000 corresponds to the just-noticeable difference threshold as predicted for solid color patches, the orange line at 2.5 E 2000 refers to an error that would be imperceptible in complex imagery and the red line at 5 E would represent errors that are likely to be objectionable.

As can be seen, the vast majority of the cases have both median and 95<sup>th</sup> %-tile differences significantly below the 1  $\Delta E$  threshold, with a single case having a 95<sup>th</sup> %-tile of around 1.5  $\Delta E$ . On average the above differences have a median of 0.4 and a 95<sup>th</sup> %-tile of 1.1 CIE  $\Delta E$ 2000 and as such are likely imperceptible. These low errors are confirmed by the feedback provided by the photographers themselves who performed the adjustments, who consistently judged the unadjusted image converted using the adjusted profile as indistinguishable from the adjusted image.



**Figure 5.** From left to right (sRGB previews): original unadjusted image in sRGB color space, unadjusted image converted to a reference printer ICC profile, adjusted image in reference printer profile space, unadjusted original image (left most above) converted to newly processed ICC profile.

Figure 5 shows an example of an image that has been processed printed first with the default printer profile (sRGB preview can be seen as the top right photo), then adjusted in Adobe Photoshop (bottom left photo) and compared against the original (top left photo) converted to the algorithmically created ICC profile (bottom right photo).

As stated earlier, it is desirable that the ICC profiles generated in this way be well behaved on other images that differ from the representative sample. The degree to which the profile behavior will have expected results will depend in part on the adjustments effected on the sample image (whether they are color-local, color- global, ...), in part on the sample image color gamut (if it is limited and a global adjustment is made, it is encoded as a local adjustment limited to the image gamut) and finally the relationship between the sample image and any other image with which the modified ICC profile is used.

To illustrate Figure 6 shows a pair of photos from the same occasion as those in Figure 5, with the unadjusted set converted to the original output profile on the left and the same unadjusted image, converted to the newly processed ICC profile created from the above example image. As can be seen the adjustments effected on the reference image (increased contrast, change of color balance, deeper blacks) transfers well to the new set of images. Note that all images have been converted to sRGB for final preview in this paper, hence it is a limited and approximate indication of their look and feel.

## Conclusions

We have presented an approach to visually adjust default ICC profiles to the individual linking of users. The ability to perform these adjustments in an environment that enables the use of intuitive tools and gives direct feedback on the effect of these means that creative users are given control over the color rendering behavior of their input or output devices.

In the course of testing this solution with professional photographers, a variety of potential uses was identified, including the compensation of inaccurate profiles, adjusting for color-casts that may arise for particular viewing conditions as well as the ability to have custom rendering choices for particular occasions or types of work (e.g. one user said they systematically have to add 5 contone values of AdobeRGB Red to their images before printing them, etc.).

Some of the benefits of the presented algorithm are the ability to perform these adjustments in a familiar environment, the processing algorithm taking into account perceptual weighting (in CIELAB space), and directly considering the interpolation scheme of the Color Management Module that applies the Look-Up Tables modified as well as the algorithm processing implicitly the image gamut. However, an important insight is also the fact that performing adjustment in this way, the adjustments of images vs the adjustments for print (or scanning) are separated. We are giving photographers the ability not to have to touch their “original” (the Adobe RGB image they have developed and retouched on-screen for hours), but instead



**Figure 6** Example photos using the reference profile (left) and adjusted profile (right) to demonstrate the consistent look-and-feel by means of the profile based adjustment.

to adjust their devices. This is a key advantage over the alternatives of doing this by means of recording Photoshop actions for example, whereby a new copy of the image is created and care has to be taken as to how to handle the new copy at the point of print. Instead using this approach, all that is needed is to chose the correct profile at the point of printing the original image, as is the case today as well.

## Acknowledgements

The authors would like to thank the Corazzini family for their kind permission to use their photographs, and the following colleagues at the Hewlett-Packard Company for their support: Carlos Amselem, Martí Maria, Óscar Martínez.

## References

- [1] Hunt R. W. G. (1970) Objectives in colour reproduction. *Journal of Photographic Science*, vol. 18, pp. 205–215.
- [2] Yule J. A. C. (1967) *Principles of Color Reproduction – Applied to Photomechanical Preproduction, Color Photography, and the Ink, Paper and Other Related Industries*, John Wiley and Sons, Inc., New York.
- [3] Woolfe G. (2007) Making Color Adjustment Accessible to Non-Experts Through the Use of Language, Fifteenth Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications, Albuquerque, New Mexico; November 2007; pp. 3-7

## Author Biography

*Peter Morovi received his Ph.D. in computing sciences from the University of East Anglia, UK in 2002 and holds a B.Sc. degree in theoretical computer science from Comenius University, Slovakia. He has been a Color Imaging Scientist at Hewlett-Packard S. L., Spain since 2007. His research interests include color vision, color reproduction, image processing and computational geometry. Dr. Morovi received the IS&T's Raymond Davis Scholarship (2001) as well as a JSPS Post-doctoral Fellowship (2005).*

*Ján Morovi received his Ph.D. in color science from the Colour & Imaging Institute of the University of Derby (UK) in 1998. After working there as a lecturer in digital color reproduction, he became Senior Color Scientist later Master Technologist at Hewlett-Packard in Barcelona, Spain and has been there since 2003. He is currently the Director the CIE's Division 8 on Image Technology and Wiley and Sons have published his book entitled 'Color Gamut Mapping.'*

*Jordi Vilar holds a B.Sc. degree in Physics from the University of Barcelona, Catalonia. He's been working as a Software Engineer at Hewlett Packard S. L., Spain since 2003. He has published over 10 journal and conference articles and his research interests include computational geometry and color science.*

*Jordi Arnabat received his M.Sc. in computer vision in 2001, holds a B.Sc. degree in physics from the University Autònoma de Barcelona, Catalonia. He's been working as Color Imaging Expert at Hewlett Packard S.L. Spain since 2008. He published over 10 journal and conference articles; his research interests include color reproduction, image segmentation, 3D reconstruction and computational geometry.*

*Ángel Albarrán received his B.Sc. degree in computer science from the Politecnia University of Barcelona, Spain, in 1996. He has been working as a Color Imaging Engineer at Hewlett Packard S.L., Spain since 2005. He is also a professional fine-art photographer with his work exhibited at numerous galleries in Spain and France and appeared in *Lens Work Extended* magazine. His research interest is focused on photography representation as a narrative and expressive medium.*