High-end colorimetric display characterization using an adaptive training set

Philippe Colantoni
Jean-Baptiste Thomas
Jon Y. Hardeberg

Abstract — A new, accurate, and technology-independent display color-characterization model is introduced. It is based on polyharmonic spline interpolation and on an optimized adaptive training data set. The establishment of this model is fully automatic and requires only a few minutes, making it efficient in a practical situation. The experimental results are very good for both the forward and inverse models. Typically, the proposed model yields an average model prediction error of about 1 ΔE_{ab}* unit or below for several displays. The maximum error is shown to be low as well.

Keywords — *Displays, color characterization, color rendering, color management, GPGPU.*

DOI # 10.1889/JSID19.8.520

1 Introduction

A cross-media color-reproduction workflow can be considered to begin with the acquisition of a color scene and end up with the display of this scene by any device or medium. It is well known that the color acquired or reproduced by different devices for the same input is not the same. Thus, the use of a color-management process is required to keep the color consistent through the entire color workflow. In a nutshell, cross-media color reproduction needs the colorimetric characterization of each color device and a color-rendering algorithm, which permits the mapping of colors from one device to another.

The colorimetric-characterization process can be defined as understanding and modeling the relationship between the input and the output, in order to control a color device. Display color characterization thus aims to model the relationship between a digital value input to the display and the resulting displayed color. A display can be considered as an interface or as a function between an input signal and a displayed color; we can consider this function and its inverse F and F^{-1} that associate the signal and the color as follows:

$$F:\begin{cases}\text{Dependent color space} \rightarrow \text{Reference color space}\\ \text{Signal} \rightarrow \text{Color} = F(\text{Signal})\end{cases}, \quad (1)$$

$$F^{-1}:\begin{cases}\text{Reference color space} \rightarrow \text{Dependent color space}\\ \text{Color} \rightarrow \text{Signal} = F(\text{Color})\end{cases}. \quad (2)$$

With these notations, Eq. (1) can be called the forward transform; meanwhile, Eq. (2) can be called the inverse or backward transform.

This article introduces a new color-characterization model based on polyharmonic splines 3-D interpolation. The novelty introduced concerns the distribution of color patches to measure for the training of the model and the freedom given to the model considering the choice of a target color space and of the kernel and smoothing factor for the interpolation. This increases noticeably the accuracy of the model. The inverse model is based on a tetrahedral interpolation, using a grid designed in RGB. As design goals, we aim for the display color-characterization model to be as accurate as possible on any type of display and we want the color correction to be done in real time (no pre-processing). Moreover, we want the model establishment not to exceed a practical time of a few minutes (the time of a coffee break).

We first present the state of the art of display color characterization in Sec. 2. We then introduce our new accurate display color-characterization model. We evaluate this method experimentally on different displays. Before concluding, we describe briefly its application to multispectral image real-time color rendering under a virtual illumination through its GPU implementation.

2 Display color characterization

In this section, we review the state of the art of display color characterization. The first part considers the methods used to model color in displays; the second part considers the inversion of color characterization models.

2.1 Modeling displays

Many color-characterization models exist; we can classify them in three groups. The first group includes physical models; the two others are empirical, *i.e.*, based on a number of measurements used to generalize the model using a mathematical process.

In the first group, one can find the models that tend to model physically the color response of the device. They are

Received 01/31/11; accepted 05/2/11.
P. Colantoni is with the Centre Interdisciplinaire d'Etudes et de Recherches sur l'Expression Contemporaine de l'Université Jean Monnet Saint-Etienne, France; e-mail: philippe.colantoni@univ-st-etienne.fr.
J-B. Thomas is with the Université de Bourgogne, Bourgogne, France.
J. Y. Hardeberg is with Gjøvik College, The Norwegian Color Research Laboratory, P.O. Box 191, Gjøvik, N-2802 Norway; telephone +47-6113-5215, e-mail: jon.hardeberg@hig.no.

often based on the assumption of the independence between the channels and of chromaticity constancy of the primaries. Then, a combination of the primary chromaticities weighted by the luminance response of the display relative to a digital input can be used to perform the colorimetric transform. These physical models are historically widely used for displays because CRT technology follows the assumptions cited above very well.[3–5] Such a model typically first aims to linearize the intensity response of the display. This can be done by establishing a model that assumes that the intensity response curve follows a mathematical function,[5–16] or by linear interpolation.[17,18] The measurement of the luminance can be done using a photometer. Some approaches propose a visual-response-curve estimation.[6,19–21] Recently, a method to retrieve the response curve of a projection display using an uncalibrated camera has been proposed.[20,22,23]

The second step of these models is commonly the use of a 3×3 matrix containing primary chromaticities to build the colorimetric transform from luminance to an additive reference color space such as *XYZ*. The primary chromaticities can be generic primaries such as sRGB[24] for some applications,[22,23] data supplied by the manufacturer,[6] or can be estimated by measurement of the device primaries at full intensity, using a colorimeter or a spectroradiometer, assuming their chromaticity constancy. In practice, this assumption does not hold perfectly. The major part of the non-constancy of primaries is coming from an offset contribution.[12] This can be corrected by applying a flare correction.[18] It is also possible to minimize the chromaticity non-constancy in finding the best chromaticity values of primaries by optimizing the components of the 3×3 matrix.[25] The colorimetric transform can as well be replaced by a 2-D interpolation in the chromaticity plane, such as in the two-step parametric model proposed by Blondé *et al.*[26]

However, the use of a physical model leads to inaccuracy due to inaccurate assumptions. An alternative approach has been derived in the masking model and modified masking model that take into account the cross-talk between channels.[27] Furthermore, the lack of chromaticity constancy can be critical, particularly for LCD technology, which has been shown to fail this assumption.[15,28] The Piecewise-Linear Assuming Variation in Chromaticity (PLVC) model[17,29–32] can be used to overcome this, specially on LCDs.[31,32] In the case of a multi-primary displays, a physical model has been introduced by Wyble *et al.*[33,34] for DLP projectors using a white segment in the color wheel, it is more common to use an empirical model.

The second group can be denoted as numerical models.[11,12,27,35,36] They are based on a training data set that typically permits the optimization of the parameters of a polynomial function to establish the transform. The numerical models suppose that the mathematical transform can be approximated by a set of equations, usually an *n*-order polynomial function. The parameters are retrieved using an *n*-order polynomial regression process based on measurements.

The last category consists of a 3-D Look-Up Table (LUT) or 3-D interpolation-based models. They are based on the measurements or estimation of a defined number of color patches. Studies assess that these methods achieve good results,[37,38] depending on the combination of the interpolation method used,[39–43] the number of patches measured and on their distribution.[38,44–47] However, to be precise enough, a high number of measurements are typically required, *i.e.*, a $10 \times 10 \times 10$ grid of patches measured in Bastani's paper.[37] Such models need high storage capacity and computational power to handle the 3-D data. The computational power is usually not a problem since graphic processor units can perform this type of task easily today. The high number of measurements needed is a greater challenge.[26] However, such models have the advantage of being technology independent, *i.e.*, no assumptions are made about the device other than that it will always have the same response at the measurement location (temporal stability). With the number of different display technologies presently on the market, and with their fast evolution, such models are the most suitable choice for stable and accurate color characterization.

2.2 Model inversion

The inversion of a display color-characterization model is of major importance for color reproduction since it provides the set of digital values to input to the device in order to display a desired color. Among the models or methods used to achieve color characterization, we will distinguish two categories.

The first one contains models that are conveniently invertible, such as the PLCC, the offset corrected PLCC and the GOG or GOGO models.[3,5,7,11,12,17,18] It is sufficient enough to invert the matrix used for the colorimetric transform and the function used to linearize the response curve to invert the model.

The second category contains the models that are not conveniently invertible for real-time application. We can list some typical problems and methods used to invert these models:

A condition or a set of conditions has to be verified, such as in the masking model.[27]

A new matrix might have to be defined by regression in numerical models.[11,12,35,36]

A full-optimization process has to be set up for each color, such as in the S-curve model II[15,16] in the modified masking model[27] or in the PLVC model.[18]

The optimization process can appear only for one step of the inversion process, as in the PLVC[17] or in the S-curve I[15,16] models.

Empirical methods based on a 3-D LUT can be inverted directly,[37] using the same geometrical structure. However, in order to obtain a better accuracy, it is common to build

another geometrical structure to yield the inverse model. For instance, it is possible to build a draft model to define a new set of color patches to be measured.[38]

It is then of use to set a 3-D LUT based on the forward model.[48] An optimization process can help to design a good LUT.

To our knowledge, the first report of creating a LUT based on the forward model is from Stokes[48] in the field of printer color characterization. In this work, the LUT is built to replace the analytical model in the forward direction. It is based on a regular grid designed in the printer CMY color space, and the same LUT is used in the inverse direction, simply by switching the domain and co-domain. Note that for displays, the forward model is usually computationally simple and that we need only to use a 3-D LUT for the inverse model. The uniform mapping of the device space leads to a non-uniform mapping in the perceptually uniform space for the inverse direction when a linear interpolation is used, and it is common to re-sample this space to create a new LUT. To do that, a new grid is usually designed in $L^*a^*b^*$ and is inverted after gamut mapping of the points located outside the gamut of the printer. Several algorithms can be used to re-distribute the data[49–51] and to fill the grid.[52–54]

Back to displays, two classical approaches using the forward model can be considered to distribute patches in the source space:

- One can directly use a regular distribution in RGB and transform it to $L^*a^*b^*$ using the forward model; this approach is the same as used by Stokes for printer characterization[48] and leads to a non-uniform mapping of the $L^*a^*b^*$ space, which leads to inaccuracy for the inverse direction. To overcome this problem, Thomas *et al.*[46,47] have proposed a method to build an optimized LUT, based on a customized RGB grid. They modified the previous framework to re-distribute the patches in RGB in order to obtain a better inverse model.
- Another approach is to distribute the patches regularly in $L^*a^*b^*$, following a given pattern or any of the methods used in printer characterization.[49–51] Then, an optimization process using the forward model can be performed for each point to find the corresponding RGB value. The major drawback of this method, compared with the previous one, is that it requires a gamut mapping algorithm to ensure that the patches are inside the gamut of the device.

The next step is generally to tetrahedrize the grid to perform a tetrahedral (or more complex) interpolation.[39] In the case of the grid designed in RGB, it is enough to tetrahedrize it directly in this space, since the structure's shape remains order while transformed to $L^*a^*b^*$.

3 Proposed model

A display color-characterization model aims to provide a function that estimates the displayed color stimuli for a given three-tuple RGB input to the display. Different approaches can be used for this purpose based on measurements of input values (*i.e.*, RGB input values to a display device) and output values (*i.e.*, *XYZ* or $L^*a^*b^*$ values measured on the screen by a colorimeter or spectrometer) (see Fig. 1).

The method we present here is based on the generalization of measurements at some positions in the color space. It is an empirical method that does not consider any assumptions based on display technology. The forward direction (RGB to $L^*a^*b^*$) is based on radial basis function (RBF) interpolation on an optimal adaptive set of measured patches. The inverse model ($L^*a^*b^*$ to RGB) is based on tetrahedral interpolation. An overview of this model is shown in Fig. 2.

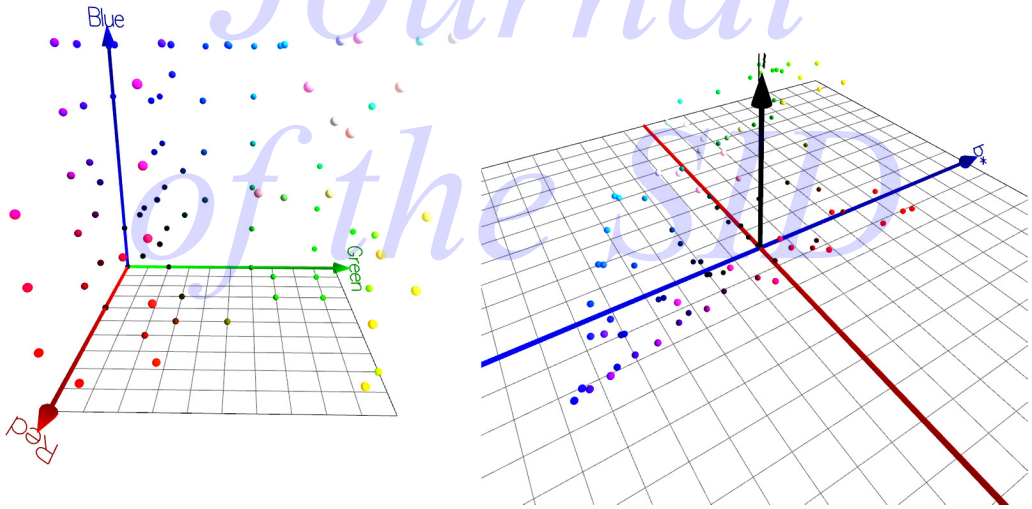

FIGURE 1 — Characterization process from RGB to $L^*a^*b^*$. The transform from RGB to $L^*a^*b^*$ is generalized to the entire color space based on some measurements and on a model.

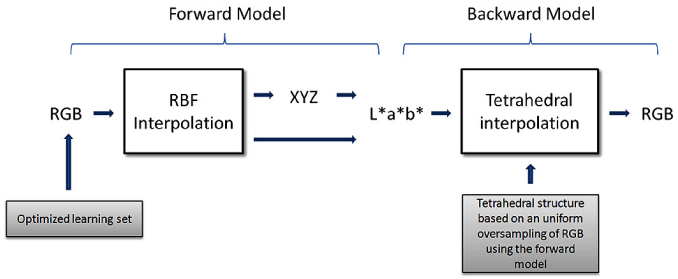

FIGURE 2 — Overview of the display color-characterization model and of the inverse model.

3.1 Forward model

Conventionally, a characterization model (or forward model) is based on an interpolation or an approximation method. We found that RBF interpolation was the most suitable model for our purpose.

3.1.1 Polyharmonic spline

Polyharmonic splines are a subset of RBF that can be used for interpolating or approximating[55,56] arbitrarily distributed data. They are defined such as:

$$\begin{cases} \phi(r) = r^k & \text{if } k = 1,3,5,\ldots \\ \phi(r) = r^k \ln(r) & \text{if } k = 2,4,6,\ldots \end{cases}.$$

In practice, in the second case, in order to avoid problems at $r = 0$, we use $\phi(r) = r^{k-1}\ln(r^r)$ when $r < 1$.

In color imaging, beside of this method and its previous version,[38,44,45] we only know the use of thin plate splines (TPS) for printer colorimetric characterization.[57] TPS are a subset of polyharmonic splines (bi-harmonic splines). Sharma and Shaw[57] recalled the mathematical framework and presented some applications and results for printer characterization. They showed that by using TPS, they achieved a better result than as using local polynomial regression. They showed that by using a smoothing factor, error in measurement impact can be avoided at the expense of the computational cost that optimize this parameter. However, they did not study the data distribution influence (but they stated that the data distribution can improve the accuracy in their conclusion), neither the use of other kernels for interpolation. The main interest of the method we propose is to take into account these parameters.

3.1.2 RBF interpolation and approximation

The idea behind RBFs is to build a function f whose graph passes through the data and minimizes a bending energy function. For a general M-dimensional case, we want to interpolate a valued function $f(X) = Y$ given by the set of values $f = (f_1, \ldots, f_N)$ at the distinct points $X = x_1, \ldots, x_N \subset \Re^M$. We choose $f(X)$ to be a RBF of the shape:

$$f(x) = p(x) + \sum_{i=1}^{N} \gamma_i \phi\left(\|x - x_i\|\right), \quad x \in \Re^M,$$

where p is a polynomial, γ_I is a real-valued weight, ϕ is a basis function, $\phi{:}\Re^M \to \Re$, and $\|x - x_i\|$ is the euclidean norm between x and x_i. Therefore, a RBF is a weighted sum of translations of a radially symmetric basis function augmented by a polynomial term. Different basis functions (kernel) $\phi(x)$ can be used.

Considering the color problem, we want to establish three three-dimensional functions $f_1(x, y, z)$. The idea is to use a sum of polynomials of small degrees instead of a high-order polynomial (such as in a classical polynomial fitting). For a set of data $\{(x_i, y_i, z_i, w_i)\}_{i=1}^{n}$ [where $w_i = f(x_i, y_i, z_i)$]) the minimizing function is given by

$$f(x,y,z) = b_0 + b_1 x + b_2 y + b_3 z + \sum_{j=1}^{n} a_j \phi\left(\left\|(x - x_j, y - y_j, z - z_j)\right\|\right), \tag{3}$$

where the coefficients a_j and $b_{0,1,2,3}$ are determined by requiring exact interpolation using the following equation

$$w_i = \sum_{j=1}^{n} \phi_{ij} a_j + b_0 + b_1 x_i + b_2 y_i + b_3 z_i \tag{4}$$

for $1 \le n$, where $\phi_{ij} = \phi[\|(x_i - x_j, y_i - y_j, z_i - z_j)\|]$. In matrix form this is

$$h = Aa + Bb, \tag{5}$$

where $A = [\phi_{ij}]$ is an $n \times n$ matrix and $\mathbf{B}$ is an $n \times 4$ matrix whose rows are $[1\ x_i\ y_i\ z_i]$. An additional requirement in order to ensure the data be not to be collinear (the non-singularity of the interpolation would not be guaranteed with our algorithm in the case of collinear data[a]) is that

$$B^T a = 0. \tag{6}$$

These two vector equations can be solved to obtain

$$a = A^{-1}(h - Bb) \text{ and } b = (B^T A^{-1} B)^{-1} B^T A^{-1} h.$$

It is possible to provide a smoothing term. In this case, the interpolation is not exact and becomes an approximation. The modification is to use the equation

$$h = (A + \lambda I)a + Bb, \tag{7}$$

$a = (A + \lambda I)^{-1}(h - Bb)$ and $b = [B^T(A + \lambda I^{-1}B)^{-1}B^T(A + \lambda I)^{-1}h$.

where $\lambda > 0$ is a smoothing parameter and I is the $n \times n$ identity matrix.

[a] Another way could be to add some jitter to the data, thus to avoid this requirement.

3.2 Free parameters of the model

3.2.1 Polyharmonic kernels

In our context, we have evaluated the use of a set of three real functions as kernel, the biharmonic [$\phi(x) = x$], triharmonic [$\phi(x) = x^3$], and thin-plate spline [$\phi(x) = x^2\ln(x)$] with x the distance from the origin. The use of a given basis function depends on the display device that is characterized and gives some freedom to the model.

3.2.2 Target color space

Our forward model uses $L^*a^*b^*$ as default target ($L^*a^*b^*$ is a target well adapted for the gamut clipping that we use). This does not imply that we have to use this space as target for the RBF interpolation. In fact, we considered two choices. We can use either $L^*a^*b^*$, which seems to be the most logical target, or XYZ associated with a XYZ to $L^*a^*b^*$ color transformation. The use of different color spaces as targets gives us another degree of freedom.

3.2.3 Smoothing factor choice

Once the kernel and the color-space target are fixed, the smoothing factor, included in the RBF interpolation model used here [see Eq. (7)] is the only parameter that can be used to change the properties of the transformation. With a zero value, the model is a pure interpolation. With a different smoothing factor, the model becomes an approximation. This is an important feature because it helps us to deal with the measurement problems due to the display temporal stability and to the repeatability of the measurement device.

3.3 Adaptively optimized learning data set

In order to increase the reliability of the model, we introduce a new way to determine the learning data set for the polyharmonic splines interpolation (*e.g.*, the set of color patches measured on the screen). We found that our interpolation model was most efficient when the learning data set used to initialize the interpolation was regularly distributed in our destination color space ($L^*a^*b^*$). This new method is based on a regular 3-D sampling of $L^*a^*b^*$ color space combined with a forward–inverse refinement process after the selection of each patch. This algorithm allows us to find the optimal set of RGB colors to measure (see Fig. 3).

This technique needs to incrementally select the RGB color patches that will be integrated into the learning database. For this reason, it has been integrated into a custom software tool that is able to drive a colorimeter or a spectroradiometer. This software also measures a set of 100 random test patches equiprobably distributed in RGB used to determine the accuracy of the model.

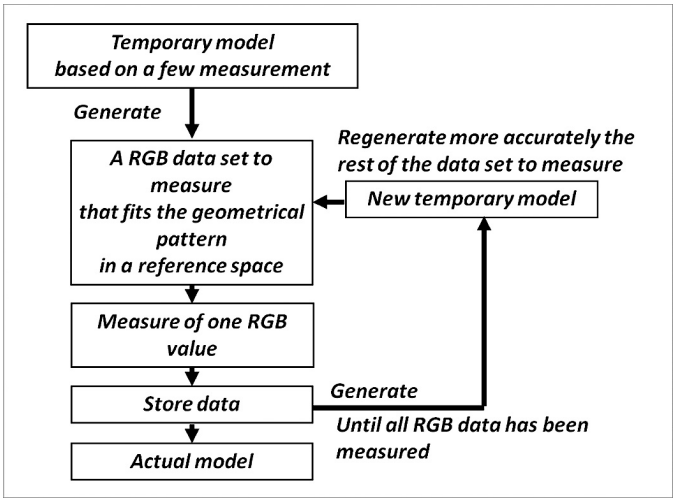

FIGURE 3 — Overview of the iterative selection of the optimal data set to measure. Every time we measure a color patch, we recomputed the rest of the data set to be measured.

3.3.1 Iterative selection of patches

In Ref. 38, the selection of the patches, in order to build the forward model, is based on an iterative process that considers the entire model. The presented algorithm considers only one iteration. First, the original set of patches is measured, and a temporary inverse model is built up. Using this temporary model, the actual forward model is based on measured patches well distributed in the destination space. Well distributed means that the patches are as equidistant as possible in the destination color space, *i.e.*, $L^*a^*b^*$ or CIEXYZ. The distribution we used in our algorithm is the same. It is based on a 3-D hexagonal grid in $L^*a^*b^*$ that is described in the following. Stauder *et al.*[38] stated that as many iteration as wanted can be used at the expense of a new measurement series at each iteration.

The approach we used in our model is different. We consider that the cost of measurement time does not fit with the reality of an application (the number of measurements for accurate display color characterization is a major problem, such as for printers, that is debated in recent literature, such as in Ref. 26 or Refs. 46 and 47). We therefore do not want to measure more than a second data set. However, the establishment and evaluation of a model is fast even on CPU. Our forward model is then based on a refinement of the model after each measurement, starting from the brighter point, ending with the darker point. Between each measurement, a new model is set up, and the RGB value (position) of the patches that have not been measured yet is re-evaluated, using a new and more accurate temporary model at each iteration. At the end, the forward model is as precise as possible considering a given number of patches. The choice to describe the grid from higher to lower luminance is defended by the commonly known fact that any measurement device is usually more accurate in higher luminance. Following this choice, we build up a more homogeneous model because the lower accuracy of the measurement device is compensated by a better estimation

of the patch to measure. This way of distribution constrains the use of this model to displays because there is a need to re-evaluate the next patch value to measure after each measurement. For instance, it would be too much time and money consumed to build such a model for a printer, even more so for a camera.

3.4 Inverse model using tetrahedral interpolation

While the forward model defines the relationship between the device color space and the CIE system of color measurement, we present in this section the inversion of this transform. Our problem is to find, for $L^*a^*b^*$ values, the corresponding RGB values (for a display device previously characterized).

This inverse model could use the same interpolation methods previously presented, but we used a new and more accurate method proposed in Ref. 44. This new method uses the fact that if our forward model is very good, then it is associated with an optimal patch database (see Sec. 3.3). Basically, we use a hybrid method; a tetrahedral linear interpolation associated with an oversampling of the RGB cube (see Fig. 4). We have chosen this interpolation method because we can generate a large number of data points, which makes the linear interpolation accurate. Moreover, the geometrical aspect of the tetrahedral structure fits well with a gamut clipping algorithm.

We built the initial tetrahedral structure by using a uniform over-sampling of the RGB cube ($n \times n \times n$ samples). This over-sampling process uses the forward model to compute the corresponding structure in $L^*a^*b^*$ color space, such as in the method proposed by Stokes.[48] Once this structure is built, we can compute, for an unknown C_{Lab} color, the associated C_{RGB} color by using tetrahedral interpolation.

The oversampling used is not based on the same number of points for each axis of RGB. It is computed according to the shape of the display device gamut in $L^*a^*b^*$ color space. Note that this concept differs from the work of Stauder *et al.*[38] that uses a regular grid in RGB, such as presented by Stokes.[48] We found that than an equivalent to $36 \times 36 \times 36$ samples was a good choice. Using such a tight structure locally linearizes our model, which becomes perfectly compatible with the use of a tetrahedral interpolation. The selection of the number of patches along each axis is performed by using a brute-force approach, such as used in Ref. 47.

4 Experimental setup and results

We want our method to work practically on most displays commonly used in museum laboratories (as well as potentially other high-end spectral and color-reproduction workflows). We decided to test our method on a relatively large set of displays: two CRTs using Diamondtron technology (Mitsubishi DiamondPro SB2070 and DiamondPlus 230) were tested as well as three LCDs: one TN technology (Sensy 24KAL), a wide gamut display using TN technology (Hewlett-Packard HP2408w) and one based on IPS technology (EIZO CG301W). The measurement device as well has to be commonly found in laboratories and should allow us to perform automatic measurements. We used a spectrophotometer EyeOne Pro from X-Rite.

We need to find the best inverse model that allows us to determine, with a maximum of accuracy, the RGB values for a given set of XYZ values. In order to complete this task we must define an accuracy criteria. We chose to multiply the average ΔE_{ab}^* by the standard deviation (STD) of ΔE_{ab}^* of the set of 100 patches evaluated with a forward model. Using the forward model makes sense because it is used to generate the inverse model data. Moreover, we combined the standard deviation of the error and the mean error in

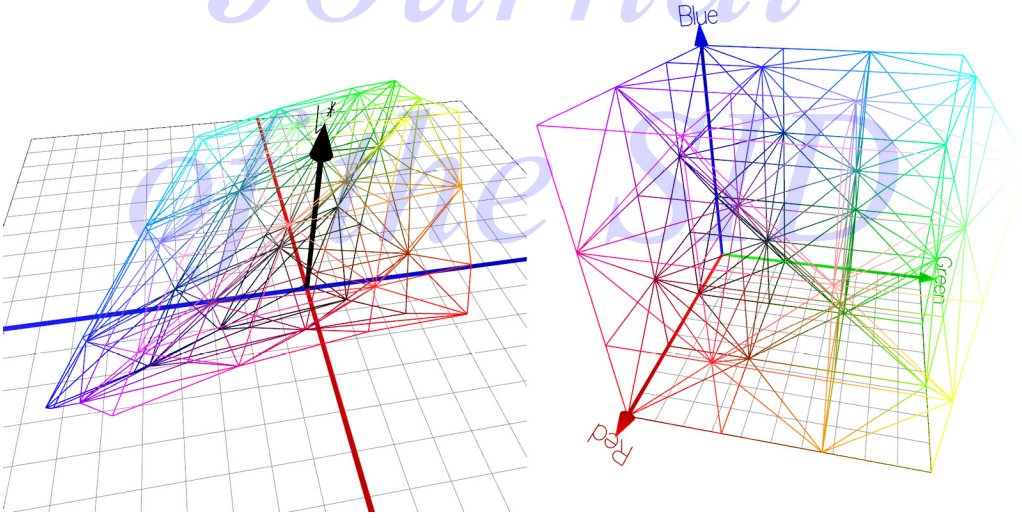

FIGURE 4 — Tetrahedral structure in $L^*a^*b^*$ and the corresponding structure in RGB.

order to have good average accuracy coupled with a small dispersion around it. For the evaluation of the inverse model, we chose to evaluate the accuracy of the transform, more than the color difference induced by the transform. For this goal, we used the Euclidean distance in the destination color space, namely, RGB. Thus, for a predicted triplet (R_1, G_1, B_1) and another (R_2, G_2, B_2) used to display the wanted color, the difference is

$$\Delta RGB = \sqrt{(R1-R2)^2+(G1-G2)^2+(B1-B2)^2}.$$

We compare our results with the same model without an adaptive selection of data, then we can clearly see the improvement generated. We do not consider a simpler model, such as the PLCC or GOGO since they have been shown to be clearly less accurate than the type of models we are evaluating, at the expense of simplicity. Moreover, it would be difficult and unfair to compare a matrix model, which requires, for instance, from eight (10 including primaries) to 24 measurements with a method that requires 216 measures and some re-computing of patches after each measure.

4.1 Measurements

Before we show our results, we should discuss color measurement. Two main factors will influence these measurements:

1. **The stability of the display device:** A display device is linked to a power supply and a light source (for LCDs, video projectors, *etc.*) or an electron gun (for CRTs, SEDs, FEDs, *etc.*) or plasma cells. We cannot expect these elements to be perfectly stable (especially with non-professional equipments). The result is a color-rendering instability.
2. The measurement devices give values with a tolerance. In the case of a colorimeter, this tolerance depends on different factors: the technology used, the brightness of the color, *etc.*

We need to know the combined influence of these two factors. In order to quantify it, we performed the following experiment for each display device tested: we send and measure 64 color patches (4 × 4 × 4) uniformly sampled patches) 30 times (20 minutes measurement).

We computed for each color patch the ΔE_{ab}* mean, max, and standard deviation. Table 1 shows the mean of these values for the 64 patches during all the measurement. Table 2 shows the result (ΔE mean and max) between two consecutive measurements.

Whatever the model used during the calibration process it will include this measurement error. We can also be confronted to a relatively instable display, even on a short period of time (with the HP2408w LCD). This type of display cannot provides reliable colors.

TABLE 1 — Display and measure repeatability.

Display Device	ΔE^*_{ab} Mean	Mean (ΔE^*_{ab} STD)	ΔE^*_{ab} Max
SB2070 - CRT Mitsubishi	**0.234267**	0.159998	1.55088
HP2408w - LCD Hewlett-Packard	1.93712	1.35595	10.8971

TABLE 2 — Repeatability between two consecutive measurements.

Display Device	ΔE^*_{ab} Mean	ΔE^*_{ab} Max
SB2070 - CRT Mitsubishi	0.102522	0.547393
HP2408w - LCD Hewlett-Packard	0.183674	0.664864

TABLE 3 — Part of the results obtained in order to evaluate the best model parameters. The presented results consider $L^*a^*b^*$ as the target color space and a triharmonic kernel for a CRT monitor SB2070 Mitsubishi DiamondPro.

Smoothing Factor	0	0.0001	0.001	0.01	0.1
ΔE Mean	0.379	0.393	**0.376**	0.386	0.739
ΔE STD	0.226	0.218	**0.201**	0.224	0.502
ΔE Max	1.374	1.327	**1.132**	1.363	2.671
ΔE 95%	0.882	0.848	0.856	**0.828**	1.769
ΔRGB Mean	**0.00396**	0.00459	0.00438	0.00421	0.00826
ΔRGB STD	**0.00252**	0.00323	0.00316	0.00296	0.00728
ΔRGB Max	0.01567	0.02071	0.01768	**0.01554**	0.05859
ΔRGB 95%	**0.00886**	0.01167	0.01162	0.01051	0.01975

TABLE 4 — Part of the results obtained in order to evaluate the best model parameters. The presented results consider *XYZ* as target color space and a triharmonic kernel for a CRT monitor SB2070 Mitsubishi DiamondPro.

Smoothing Factor	0	0.0001	0.001	0.01	0.1
ΔE Mean	0.495	0.639	0.539	**0.332**	0.616
ΔE STD	0.293	0.424	0.360	**0.179**	0.691
ΔE Max	1.991	2.931	2.548	**1.075**	4.537
ΔE 95%	1.000	1.427	1.383	**0.7021**	1.751
ΔRGB Mean	0.00674	0.00905	0.00720	**0.00332**	0.00552
ΔRGB STD	0.00542	0.00740	0.00553	**0.00220**	0.00610
ΔRGB Max	0.02984	0.03954	0.03141	**0.01438**	0.04036
ΔRGB 95%	0.01545	0.02081	0.01642	**0.00597**	0.01907

4.2 Optimal model

The selection of the optimal parameters can be done using a brute-force method. We compute for each kernels (*i.e.*, biharmonic, triharmonic, thin-plate spline), each color space target (*L*a*b**, *XYZ*) and several smoothing factors (0, *i.e.*, 005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, and 0.1) the values of this criteria and we select the minimum. For example, Tables 3 and 4 show the results obtained for a SB2070 Mitsubishi DiamondPro with a triharmonic kernel for *L*a*b** (Table 2) and *XYZ* (Table 3) as target color space (using a learning data set of 216 patches):

According to our criterion, the best kernel is the triharmonic with a smoothing factor of 0.01 and *XYZ* as target space.

The measurement process took about 5 minutes and the optimization process took 1 minute (with a four-core processor, Intel core2 quad 2.8 GHz). We reached our goal which was to provide an optimal model during the time of a coffee break for a user.

Our different experimentation showed that a 216 patches learning set was a good compromise (equivalent to a $6 \times 6 \times 6$ sampling of the RGB cube). A smaller data set gives us a degraded accuracy, a larger gives us similar results because we are facing the measurement problems introduced previously.

4.3 Optimized learning data set

Tables 4 and 5 show the results obtained with our model for two displays of different technologies. These tables show clearly how the optimized learning data set can produce better results with the same number of patches.

4.4 Results on different displays

The following table presents different results obtained for three other displays (two LCDs and one CRT).

We can see here that our model gives very good results on a wide range of displays.

4.1 GPU-based implementation and real-time application

This color-management method is based on a conversion process that will compute for *XYZ* values the corresponding RGB values. Often a colorimetric characterization model

TABLE 5 — Accuracy of the model established with 216 patches in forward and inverse direction for a LCD wide-gamut display (HP2408w). The distribution of the patches plays a major role in the model accuracy.

	Forward Model		Inverse Model	
	ΔE Mean	ΔE Max	ΔRGB Mean	ΔRGB Max
Optimized	**1.057**	4.985	**0.01504**	0.1257
Uniform	1.313	9.017	0.01730	0.1168

TABLE 6 — Accuracy of the model established with 216 patches in forward and inverse direction for a CRT display (Mitsubishi SB2070). The distribution of the patches plays a major role in the model accuracy.

	Forward Model		Inverse Model	
	ΔE Mean	ΔE Max	ΔRGB Mean	ΔRGB Max
Optimized	**0.332**	1.075	**0.00311**	0.01267
Uniform	0.435	1.613	0.00446	0.01332

TABLE 7 — Accuracy of the model for other displays. We can notice that we obtain very good results for many devices.

	Forward Model		Inverse Model	
	ΔE Mean	ΔE Max	ΔRGB Mean	ΔRGB Max
EIZO CG301W (LCD)	0.783	1.906	0.00573	0.01385
Sensy 24KAL (LCD)	0.956	2.734	0.01308	0.06051
DiamondPlus 230 (CRT)	0.458	2.151	0.00909	0.06380

can be shown to be unusable in practice, due to a too complicated computation for real-time application.

In order to demonstrate the practical possibility of use of this algorithm, we implemented and used this method with success within a multispectral application. By using multispectral images and a tool to define a virtual illuminant in real time, we were able to build the corresponding CIEXYZ image. From these CIEXYZ data and the reference white of the display, we can compute CIELAB values and using our model, we computed the RGB data that would permit us to precisely visualize the colorimetric data of the image.

We managed to develop a practical real-time process implemented in GPU-based software.

It is possible to implement the presented algorithm with a specific GPU language, such as CUDA, but the software will only work with a CUDA-compatible GPU (*n*vidia G80, G90, and GT200). The goal was to create a working application on a large number of GPUs (AMD and *n*vidia™ GPUs); for this reason we chose to implement a classical method using a 3-D LUT used by OpenGL 2.0 shaders.

During an initialization process, we build a three-dimensional RGBA floating-point texture that covers the *L*a*b** color space. The alpha channel of the RGBA values saves the distance between the initial *L*a*b** value and *L*a*b** value obtained after the gamut-mapping process. If this value is 0, the *L*a*b** color that has to be converted is in the gamut of the display, otherwise this color is out of the gamut and the closest color according to a given gamut-mapping process is displayed. This allows color errors due to the screen's inability to display every visible color in real time.

Finally, the complete color pipeline includes a reflectance to *XYZ* conversion (considering an illuminant given by the user), then an *XYZ* to *L*a*b** conversion (using the white of the screen as reference), our colorimetric characterization process, and a gamut-mapping algorithm.

5 Conclusion

We proposed a new color display-characterization model that consists of an optimal combination of measured samples and interpolation settings. We are able to predict with a high accuracy the color displayed using a few measurements. For instance, we obtained an average error below 1 ΔE_{ab}* for most tested displays. In most cases, the model error is close to the repeatability error of the characterized device. We have shown that it is possible to implement an accurate color-management process even for a real-time color reconstruction.

This color-management process is based only on colorimetric considerations. The next step is to introduce a color-appearance model in the color workflow. By now, this method has been used within a custom color-management module. However, it is possible to standardize the output of the process using a 3-D LUT embedded in an ICC profile.

Acknowledgment

The authors would like to thank Thomson Corporate Research and the Laboratoire d'Informatique Graphique et d'Ingénierie de la vision (LIGIV), which initiated this work. The application to multispectral images of art paintings is supported by the Centre de Recherche et de Restauration des Musées de France (C2RMF).

References

1 P. Colantoni and J.-B. Thomas, "A color management process for real time color reconstruction of multispectral images," *Lecture Notes in Computer Science*, A.-B. Salberg, J. Y. Hardeberg, and R. Jenssen (eds.), **5575**, 128–137 (2009).

2 J.-B. Thomas, "Tatouage d'image couleur, en vue de l'insertion d'une palette de couleurs caractéristiques des couleurs d'une image," *Master Thesis* (2006).

3 W. Cowan and N. Rowell, "On the gun independency and phosphor constancy of color video monitor," *Color Res. & Application* **11**, S34–S38 (1986).

4 D. H. Brainard, "Calibration of a computer-controlled color monitor," *Color Res. & Application* **14**, 23–34 (1989).

5 R. S. Berns *et al.*, "CRT colorimetry. Part II: Metrology," *Color Res. & Application* **18**, No. 5, 315–325 (1993).

6 W. B. Cowan, "An inexpensive scheme for calibration of a colour monitor in terms of CIE standard coordinates," *SIGGRAPH Comput. Graph* **17**, No. 3, 315–321 (1983).

7 R. S. Berns *et al.*, "CRT colorimetry. Part I: Metrology," *Color Res. & Application* **18**, No. 5, 299314 (1993).

8 G. Sharma, *Digital Color Imaging Handbook* (CRC Press, 2003).

9 R. Berns, "Methods for characterizing CRT displays," *Displays* **16**, No. 4, 173–182 (1996).

10 CIE, *The Relationship between Digital and Colorimetric Data for Computer-Controlled CRT Displays*, CIE, Publ. 122 (1996).

11 N. Katoh *et al.*, "An accurate characterization of CRT monitor (II) proposal for an extension to CIE method and its verification," *Opt. Rev.* **8**, No. 5, 397–408 (2001).

12 N. Katoh *et al.*, "An accurate characterization of CRT monitor (I) verification of past studies and clarifications of gamma," *Opt. Rev.* **8**, No. 5, 305–314 (2001).

13 IEC:61966-3, "*Color measurement and management in multimedia systems and equipment, Part 3: Equipment using CRT displays*," IEC (1999).
14 Y. Yoshida and Y. Yamamoto, "Color calibration of LCDs," *Proc. 10th Color Imaging Conference,* 305–311 (2002).
15 Y. Kwak and L. MacDonald, "Characterisation of a desktop LCD projector," *Displays* **21**, No. 5, 179–194 (2000).
16 Y. Kwak *et al.,* "Controling color of liquid-crystal displays," *J. Soc. Info. Display* **11**, No. 2, 341–348 (2003).
17 D. L. Post and C. S. Calhoun, "An evaluation of methods for producing desired colors on CRT monitors," *Color Res. & Application* **14**, 172–186 (1989).
18 L. Jimenez Del Barco *et al.,* "Considerations on the calibration of color displays assuming constant channel chromaticity," *Color Res. & Application* **20**, 377–387 (1995).
19 A. Neumann *et al.,* "Interactive perception based model for characterization of display device," *SPIE Proc. Color Imaging IX: Processing, Hardcopy, and Applications IX* **5293** 232–241 (Dec. 2003).
20 E. B. Mikalsen *et al.,* "Verification and extension of a camera-based end-user calibration method for projection displays," *CGIV,* 575–579 (June 2008).
21 R. Klassen *et al.,* "Visually determining gamma for softcopy display," *Proc. 13th Color Imaging Conference,* 234–238 (2005).
22 R. Bala and K. Braun, "A camera-based method for calibrating projection color displays," *Proc. 14th Color Imaging Conf.* 148–152 (2006).
23 R. Bala *et al.,* "Efficient and simple methods for display tone-response characterization," *J. Soc. Info. Display* **15**, No. 11, 947–957 (2007).
24 M. Anderson *et al.,* "Proposal for a standard default color space for the internet: srgb," *Proc. 4th Color Imaging Conference,* 238–245 (1995).
25 E. A. Day *et al.,* "Colorimetric characterization of a computer-controlled liquid crystal display," *Color Res. & Application* **29**, No. 5, 365–373 (2004).
26 L. Blond'e *et al.,* "Inverse display characterization: A two-step parametric model for digital displays," *J. Soc. Info. Display* **17**, No. 1, 13–21 (2009).
27 N. Tamura *et al.,* "Masking model for accurate colorimetric characterization of LCD," *J. Soc. Info. Display* **11**, No. 2, 333–339 (2003).
28 D. H. Brainard *et al., Display Characterization, Encyclopedia of Imaging Science and Technology* (Wiley, New York, 2002).
29 W. W. Farley and J. C. Gutmann, "Digital image processing systems and an approach to the display of colors of specified chrominance," Technical Report HFL-80-2/ONR-80, Virginia Polytechnic Institute and State University, Blacksburg, VA (1980).
30 D. L. Post and C. S. Calhoun, "Further evaluation of methods for producing desired colors on CRT monitors," *Color Res. & Application* **25**, 90–104 (2000).
31 J.-B. Thomas *et al.,* "Additivity based LC display color characterization," *Proc. Gjøvik Color Imaging Symposium,* J. Y. Hardeberg *et al.* (eds.), **3**, 50–55 (2007).
32 J.-B. Thomas *et al.,* "The PLVC color characterization model revisited," *Color Res. & Application* **33**, No. 6, 449–460 (2008).
33 D. R. Wyble and H. Zhang, "Colorimetric characterization model for dlp projectors," *Proc. 11th Color Imaging Conference,* 346–350 (2003).
34 D. R. Wyble and M. R. Rosen, "Color management of dlp projectors," *Proc. 12th Color Imaging Conference,* 228–232 (2004).
35 P. Green and L. MacDonald (eds.), *Color Engineering* (Wiley, Chichester, U.K., 2002).
36 S. Wen and R. Wu, "Two-primary crosstalk model for characterizing liquid crystal displays," *Color Res. & Application* **31**, No. 2, 102–108 (2006).
37 B. Bastani *et al.,* "Calibrated color mapping between LCD and CRT displays: A case study," *Color Res. & Application* **30**, No. 6, 438–447 (2005).
38 J. Stauder *et al.,* "Device, system and method for characterizing a colour device," European Patent WO/2007/116077 (October 2007).
39 L. M. Kasson *et al.,* "Performing color space conversions with three-dimensional linear interpolation," *J. Electron. Imaging* **4**(3), 226–250 (1995).
40 I. Amidror, "Scattered data interpolation methods for electronic imaging systems: A survey," *J. Electron. Imaging* **11**, No. 2, 157–176 (2002).
41 F. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence* **11**, No. 6, 567–585 (1989).
42 G. M. Nielson *et al.* (eds.), *Scientific Visualization, Overviews, Methodologies, and Techniques* (IEEE Computer Society, 1997).
43 H. Akima, "A new method of interpolation and smooth curve fitting based on local procedures," *J. Association for Computing Machinery* **17**, 589–602 (1970).
44 P. Colantoni *et al.,* "Device and method for characterizing a colour device" (2005).
45 J. F. Stauder *et al.,* "Device and method for characterizing a colour device" (September 2006).
46 J.-B. Thomas *et al.,* "An inverse display color characterization model based on an optimized geometrical structure," in *Proc. SPIE,* R. Eschbach *et al.* (eds.), **6807**, 1–12 (2008).
47 J.-B. Thomas *et al.,* "A geometrical approach for inverting display color-characterization models," *J. Soc. Info. Display* **16**, No. 10, 1021–1031 (2008).
48 M. Stokes, "Method and system for analytic generation of multi-dimensional color lookup tables" (March 1997).
49 J. Z. Chan *et al.,* "Sequential linear interpolation of multidimensional functions," *IEEE Trans. Image Processing* **6**, 1231–1245 (Sept. 1997).
50 R. E. Groff *et al.,* "Piecewise linear homeomorphisms: The scalar case," *IJCNN* (3), 259–264 (2000).
51 S. Dianat *et al.,* "Dynamic optimization algorithm for generating inverse printer map with reduced measurements," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing* (May 2006).
52 D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proc. 23rd ACM National Conference,* 517–524 (1968).
53 R. Balasubramanian and M. S. Maltz, "Refinement of printer transformations using weighted regression," *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series,* J. Bares (ed.), **2658**, 334–340 (Mar. 1996).
54 D. E. Viassolo *et al.,* "Practical algorithm for the inversion of an experimental input-output color map for color correction," *Opt. Eng.* **42**, No. 3, 625–631 (2003).
55 J. C. Carr *et al.,* "Reconstruction and representation of 3d objects with radial basis functions," *Proc. 28th Annual SIGGRAPH Conference on Computer Graphics and Interactive Techniques,* 67–76 (2001).
56 J. Duchon, "Splines minimizing rotation-invariant seminorms in sobolev spaces," *Lecture Notes in Mathematics,* No. 571, 85–100 (1977).
57 G. Sharma and M. Q. Shaw, "Thin-plate splines for printer data interpolation," *Proc. European Signal Proc. Conf.* (September 2006) (CDROM).
58 A. Ribés *et al.,* "Calibration and spectral reconstruction for crisatel: An art painting multispectral acquisition system," *J. Imaging Sci. Technol.* **49**, 563–573 (2005).
59 P. Colantoni *et al.,* "Gpu spectral viewer: analysing paintings from a colorimetric perspective," *Proc. 8th Intl. Symposium on Virtual Reality, Archaeology and Cultural Heritage* (Nov. 2007).
60 http://www.gpgpu.org, "General-purpose computation on graphics hardware" (2009).
61 P. Colantoni *et al.,* "Fast and accurate color images processing using 3D graphics cards," *Vision, Modeling and Visualization Workshop,* 383–390 (2003).
62 G. Wyszecki, "A regular rhombohedral lattice sampling of munsell renotation space," *J. Opt. Soc. Am.* **44**, No. 9, 725–734 (1954).
63 J.-B. Thomas and A. Trémeau, "A gamut preserving color image quantization," *Proc. 14th Intl. Conf. Image Analysis and Processing,* 221–226 (2007).
64 J.-B. Thomas *et al.,* "Image watermarking based on a color quantization process," *Multimedia Content Access: Algorithms and Systems* **6506**, No. 1, 650603 (2007).

Appendix

Sampling algorithm

The CIELAB sampling we used has already been used in different works, such as Refs. 1, 2, 45, 38, 62–64. We propose to detail this algorithm in this appendix.

We distributed samples as if they were the center of spheres in a closed packing of spheres problem. We sampled a cube including the gamut of the display in CIELAB color space by using a hexagonal closed packing.

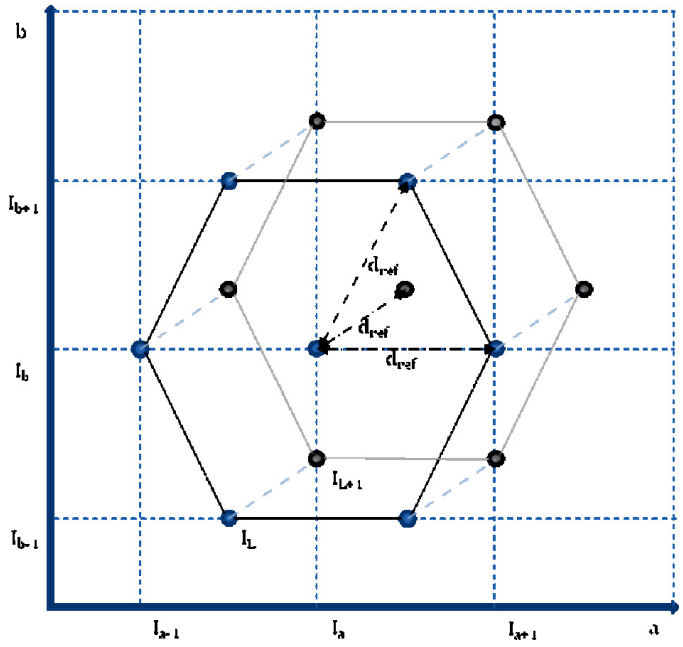

FIGURE A1 — Sampling of the $L^*a^*b^*$ color space.

Figure A1 shows the sampling algorithm as it has been designed in Refs. 45 and 63.

Let us introduce the following notations:

Let us call L_x^*, a_x^*, b_x^* the coordinates of a given color in the $L^*a^*b^*$ color space and L_y^*, a_y^*, b_y^* the coordinates of a second color. The CIE ΔE_{ab}^* color distance between these two colors is the euclidean distance, such as

$$\Delta E_{ab}^* = \sqrt{d_L^2 + d_a^2 + d_b^2}$$

with $\left|L_x^* - L_y^*\right| = d_L$, $\left|a_x^* - a_y^*\right| = d_a$, $\left|b_x^* - b_y^*\right| = d_b$.

Let d_{ref} be an arbitrary distance in L^*a^*b color space. If we consider $d_a = d_{ref}$, $d_b = 0$, and $d_L = 0$, then $\sqrt{d_L^2 + d_a^2 + d_b^2} = d_{ref}$. Likewise, if we consider $d_a = \frac{1}{2} \times d_{ref}$, $d_b = \sqrt{\frac{3}{4}} \times d_{ref}$, and $d_L = 0$, then $\sqrt{d_L^2 + d_a^2 + d_b^2} = d_{ref}$. Finally, if we consider $d_a = \frac{1}{2} \times d_{ref}$, $d_b = \frac{1}{2\sqrt{3}} \times d_{ref}$ and $d_L = \sqrt{\frac{2}{3}} \times d_{ref}$, then $\sqrt{d_L^2 + d_a^2 + d_b^2} = d_{ref}$.

Considering, now, the uniform color-space sampling, let us give L_{min}, L_{max}, a_{min}, a_{max}, b_{min}, and b_{max} the lower and upper color values of the CIELAB color space along the L^*, a^*, and b^* axis.

Considering the arrangement explained above, the 3-D grid is defined such that:

- $\left|a_{i_a}^* - a_{i_{a+1}}^*\right| = d_{ref}$ is the distance that separates two consecutive samples along the a^* axis, such that the distance that separates two samples along this axis is $\sqrt{\left(a_{i_a}^* - a_{i_{a+1}}^*\right)^2} = d_{ref}$.
- $\left|a_{i_a,i_b}^* - a_{i_a,i_{b+1}}^*\right| = \frac{1}{2} \times d_{ref}$ and $\left|b_{i_a,i_b}^* - b_{i_a,i_{b+1}}^*\right| = \sqrt{\frac{3}{4}} \times d_{ref}$ are the distances that separate two adjacent samples along the a^* and b^* axis, such that the distance that separates two samples in the a^*b^* plane is $\sqrt{\left(a_{i_a,i_b}^* - a_{i_a,i_{b+1}}^*\right)^2 + \left(b_{i_a,i_b}^* - b_{i_a,i_{b+1}}^*\right)^2} = d_{ref}$.
- $\left|a_{i_L,i_a,i_b}^* - a_{i_{L+1},i_a,i_b}^*\right| = \frac{1}{2} \times d_{ref}$, $\left|b_{i_L,i_a,i_b}^* - b_{i_{L+1},i_a,i_b}^*\right| = \frac{1}{2\sqrt{3}} \times d_{ref}$, and $\left|L_{i_L,i_a,i_b}^* - L_{i_{L+1},i_a,i_b}^*\right| = \sqrt{\frac{2}{3}} \times d_{ref}$ are the distances that separate two adjacent samples along the a^*, b^*, and L^* axis such that the distance that separates these two samples in $L^*a^*b^*$ color space is $\sqrt{\left(a_{i_L,i_a,i_b}^* - a_{i_{L+1},i_a,i_b}^*\right)^2 + \left(b_{i_L,i_a,i_b}^* - b_{i_{L+1},i_a,i_b}^*\right)^2 + \left(L_{i_L,i_a,i_b}^* - L_{i_{L+1},i_a,i_b}^*\right)^2} = d_{ref}$.
- The smaller d_{ref} is, the finer the sampling of the color space is, then the number of samples increases inversely proportionally to the cube of the distance d_{ref}.