

Tool for 3D Visualization of Color Gamuts and Volume Calculation

Cédric Marchessoux, Jürgen Jung and Geert Deroover
Agfa-Gevaert N.V.
Mortsel, Belgium

Abstract

This work presents an interactive tool for 3-dim visualization of color gamuts and calculation of their volume in CIE Lab space. In a first step, a color generator module simulates the colors obtained from a given set of dyes CMYK (RGB) in an idealized printing process. The dyes are characterized by their spectral densities as measured in solution. Color mixing is applied according to Lambert Beer law, i.e. linear combinations of the spectral densities are calculated. These represent “inner spectra” of the receiving layer and the non-linear Saunderson model is applied to take the optical effect of the air/receiver interface into account, finally leading to “external spectra”. By making complete abstraction from a halftone structure, the potential gamut volume of a given set of CMYK (RGB) inks can be benchmarked without influence or limitation by the underlying printing process or the receiving layer. A recently introduced, fast and robust 3-dim surface reconstruction method (Tight Cocone) is then used to obtain a closed surface triangulation, from which the volume of the gamut is obtained with standard methods of vector analysis. Some applications of the tool are shown in comparison with other methods.

Introduction

In the field of inkjet printing, several methods were developed to estimate and construct a potential color gamut for a dye ink set.¹⁻⁴

In this paper, we propose a new tool, which is new in the fact of grouping several methods together. The method for the determination of the potential color gamut of a dye ink set is based on a simulation of the color gamut of an imaging material in reflection geometry. This simulation can be thought of being produced by a given set of dyes in an idealized printing process. This ideal printing process is characterized by assuming:

- The validity of the Lambert-Beer law for the mixing of colorants within the image receiving layer (inner spectra).
- The applicability of the Saunderson equation in order to describe the effect of the air/receiver interface (external spectra).

The ink set consists at least of the 3 primary colors Y, M and C and an additional black component. Furthermore, the effect of extending the ink sets by

additionally introducing separate inks B, G and R for the secondary color channels, can be taken into account, thus allowing interactive simulation of hifi color systems. Also the effect of the addition of light inks can be calculated (typical dye ink sets comprise light C and light M inks). In a second step of the calculation, a surface triangulation method is used to calculate the volume of the simulated color gamut.

The choice of the ideal printing process allows to make abstraction from the real limitations of paper, printers and/or printer driver settings in any real printing process, such that two dye ink sets can objectively be compared with respect to their potential color gamut based on their spectral properties only.

Color Generator: The “Ideal Printing Process”

The model of the ideal printing process is based on the following assumptions:

Homogeneous Distribution of Colorants

Homogenous distribution of colorants (and their mixtures) in the imaging layer, i.e. continuous-tone printing like in chromogenic silverhalide photographic paper. The halftone structure of the printing process is thus not taken into account: All simulated test-patches are assumed to represent homogenous flat-fields. In addition, light-scattering processes are assumed to be absent within the image receiving layer, unless the ideal diffuse reflection due to the paper base.

Lambert-Beer Law

Mixtures and variations of the concentration of colorants within the imaging layer (inner spectra) are calculated according to the Lambert-Beer law, i.e. linear combination of the colorants absorption spectra in terms of spectral optical density $D_c(\lambda)$, $D_M(\lambda)$, $D_Y(\lambda)$, $D_K(\lambda)$ of C, M, Y and K, respectively, which are initially measured by a spectrophotometer in transmission mode (i.e. liquid solutions of the neat colorants in a cuvette with a given nominal concentration). The coefficients c , m and y (0 – 100%) represent a relative part of the nominal concentration of the colorants and run over all combinations representing the surface of the CMY cube (i.e. at least one of the coefficients c, m, y has the value 0% or 100%).

$$D(\lambda) = c.D_c(\lambda) + m.D_M(\lambda) + y.D_Y(\lambda) + k.D_K(\lambda) \quad (1)$$

$c, m, y, k \in [0-100\%]$

Gray Component Removal

In addition, gray component removal is applied, i.e. the component $k(D_c(\lambda)+D_m(\lambda)+D_y(\lambda))$ is replaced by the corresponding amount of $k D_k(\lambda)$. k denotes the minimum of the coefficients c, m, y .

$$D(\lambda)=(c-k).D_c(\lambda)+(m-k).D_m(\lambda)+(y-k).D_y(\lambda)+k.D_k(\lambda),$$

$$k=\min(c,m,y) \quad (2)$$

Secondary Component Removal

Furthermore, up to three additional inks $D_b(\lambda)$, $D_g(\lambda)$, $D_r(\lambda)$ can be introduced in the simulation for replacing the mixtures M+C, C+Y and Y+M by the corresponding secondary colors B, G and R, respectively. In case these secondary inks are used, secondary component removal is applied in addition to gray component removal. Thence, the binary components $b(D_m(\lambda)+D_c(\lambda))$, $g(D_c(\lambda)+D_y(\lambda))$ and $r(D_y(\lambda)+D_m(\lambda))$ are replaced by the corresponding amount of $b.D_b(\lambda)$, $g.D_g(\lambda)$ and $r.D_r(\lambda)$, respectively. The coefficients b, g, r denote the minimum of c, m, y in one of the corresponding binary mixtures (which are left after gray component removal), i.e. only one coefficient b, g, r differs from 0 for a given initial combination of CMYK.

$$D(\lambda)=(c-k-b-r).D_c(\lambda)+(m-k-g-r).D_m(\lambda)+(y-k-b-g).$$

$$D_y(\lambda)+k.D_k(\lambda)+r.D_r(\lambda)+g.D_g(\lambda)+b.D_b(\lambda),$$

$$k=\min(c,m,y):$$

Gray Component Removal

$$b=\min(m-k,c-k); g=\min(c-k,y-k); r=\min(y-k,m-k):$$

Secondary Component Removal

(3)

Saunderson Effect

Starting from the so calculated “internal optical density spectra” $D(\lambda)$ of colorant mixtures, that are assumed to be homogeneously distributed within the image receiving layer, the optical effect of the air / receiver interface has to be taken into account in order to realistically simulate reflection image behavior (i.e. “external spectra”). The presence of the air interface introduces two phenomena, namely i) direct external surface reflections of the incoming light beam (reflectance factor r_s) and ii) internal surface reflections (reflectance factor r_i) of light, that has been diffusely reflected by the substrate at the lower boundary of the image receiving layer. The substrate is described by its reflectance spectrum $R_s(\lambda)$ or its optical density spectrum $D_s(\lambda)$, respectively. The internal reflection at the air interface gives rise to multiple optical reflections within the imaging layer. All these effects are taken into account by the formalism of Saunderson,⁵ that therefore allows to calculate external reflection spectra $R_{ext}(\lambda)$ from internal reflection spectra $R_{int}(\lambda)$:

$$R_{ext}(\lambda)=r_s+\frac{(1-r_s)(1-r_i)R_{int}(\lambda)}{1-r_i R_{int}(\lambda)}=\frac{(1-r_s)(1-r_i)}{R_{int}^{-1}(\lambda)-r_i}$$

$$R_{int}(\lambda)=R_s(\lambda)\cdot 10^{-epi\cdot D(\lambda)}=10^{-epi(D(\lambda)+D_s(\lambda))} \quad (3)$$

The effective pathlength epi describes the factor, by which the pathlength of the light inside the imaging layer

is increased in comparison to the transmission case. This is due to geometric reasons: light is travelling twice through the layer because of reflection geometry. Furthermore, the pathlength is increased for beams other than normal with respect to the interface, e.g. in case of diffuse illumination, another factor 2 is found for the epi .

It is assumed, that r_s , r_i and epi do not vary with the wavelength λ . The parameter settings used are:

$$r_s = 0.001 \text{ (high gloss surface),}$$

$$r_i = 0.6, \text{ } epi = 2.2 \text{ (assuming 45/0 reflection geometry).}$$

For an ideal printing process a substrate without absorption is considered, i.e. $R_s(\lambda) = 1$ or $D_s(\lambda) = 0$, respectively. Be aware, that $R_s(\lambda)$ or $D_s(\lambda)$ denote inner spectra, which can be obtained from external measurements by the inverse of the Saunderson formula.

Color Space Representation

From the set of reflection spectra, that represent all the combinations of colorants on the surface of the CMY cube, the corresponding set of CIE L*a*b* coordinates are calculated based on the CIE 1976 (2 degree) observer and D50 as illuminant. Alternative observer and illuminant settings are also foreseen.

To obtain a more realistic volume of an ink set, the spectral absorption of the real paper substrate can be taken into account, i.e. $D_s(\lambda) \neq 0$. This leads to a degradation of the gamut volume. Colorimetrically this corresponds to switch white point calibration: With paper being used, the absolute L*a*b* values of the paper will be referred to as “white”, whereas relative colorimetric values are obtained without the effect of the paper base, i.e. L*a*b*(white) = (100, 0, 0).

Tools

These routines are implemented in terms of a C/C++ program. The diagram given by the figure 1 shows the workflow within the tool. An easy interface is developed in Tcl/Tk.⁶ The Visualization Toolkit (Vtk) is used for 3D's visualization.⁷

Vtk consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java and Python. Vtk can be implemented on nearly every Unix-based platform, PC's (Windows 95/98/NT/2000/XP)... The design and implementation of the library has been strongly influenced by object-oriented principles. The graphics model in Vtk reflects a higher level of abstraction than rendering libraries like OpenGL. This means it is much easier to create useful graphics and visualization applications. In Vtk applications can be written directly in C++, Tcl, Java, or Python. In fact, using the interpreted languages Tcl or Python with Tk, and even Java with its GUI class libraries, it is possible to build useful applications considerably fast.

Tcl (Tool Command Language) is a scripting language with a minimal structure and syntax. The use of a scripting language for Vtk must be seen as a trade-off between performance and quick programming without

many overheads. For small, simple programs Tcl is a very useful tool. Another strength of Tcl is the Tk (Tool Kit) package, which provides a graphical interface development tool. The Tk package can be easily used with several Tcl commands to place and manipulate widgets, bind events (such as keyboard or button events) to Tcl commands and manipulate selection and focus. The elements facilitate to build interfaces in Tcl/Tk.

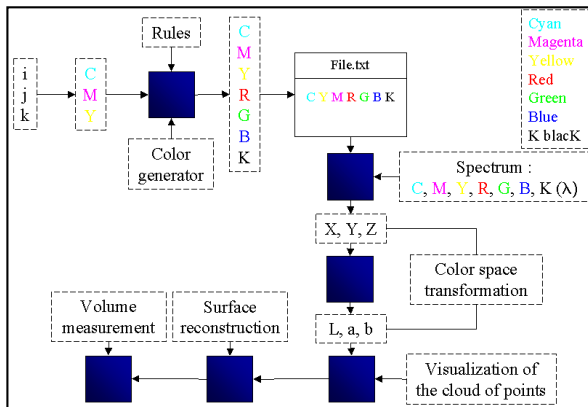


Figure 1. Workflow diagram

Gamut Visualization and Volume Measurement

The set of CIE $L^*a^*b^*$ color values represents an ordered cloud of points of the surface of the potential color gamut. These can be used for a 3-dim visualization and the determination of the corresponding gamut volume.

To calculate the volume enclosed by these surface points, first a surface triangulation technique is applied. From these, a set of triangle facets is obtained, which completely covers the color gamut. This is a non-trivial task for the shape of objects encountered in gamut characterization, namely sharp cusps and edges. Standard triangulation methods typically encounter problems at the border lines of the separate sheets, namely non-closed surface edges. This artifact cannot simply be overcome in a general and robust way (i.e. useful for all kind of gamut shapes).

Therefore, an improved version of the Cocone algorithm.⁸ "Tight Cocone",⁹ has been selected to obtain a reasonably robust surface representation for such 3-dim objects exhibiting cusps and edges. This method uses an approximation of the medial axis from the Voronoi diagram in a scale and density independent manner. The assumption behind this method is that the medial axis of a shape provides a compact representation of its features and their connectivity. The underlying algorithm approximates the medial axis in a scale and density independent manner such that convergence is guaranteed.

With a surface triangulation at hands, a list of coordinates representing all triangles on the surface is available. Furthermore, an initially arbitrarily chosen, but then fixed point inside the cloud of points, e.g. the center of gravity in terms of $L^*a^*b^*$, is taken as "inner point".

Together with this "inner point" each surface triangle facet forms a tetrahedron, from which the volume can be calculated in case of a simple convex surface hull. This approach uses standard methods of vector analysis (cf equation 4). In this equation, \mathbf{a} , \mathbf{b} and \mathbf{c} denote the vectors connecting the aforementioned "inner point" with each of the 3 corner points of a surface triangle facet. The total volume of the color gamut is then obtained by summing up the volume of all individual tetrahedron (cf figure 2).

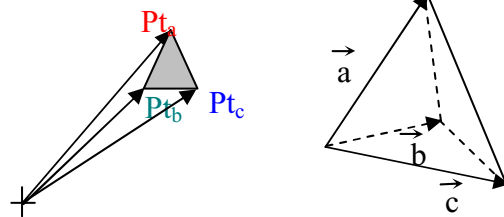


Figure 2. Tetrahedron for the volume measurement

$$V = \frac{1}{6} \vec{a} \cdot \vec{b} \times \vec{c} \tag{4}$$

Final Interface

Figure 4 shows a basic interface in order to calculate the gamut volume of a hypothetical set of inks and application of Gray Component Removal. The interface is organized in a main window and two secondary windows. The user has to follow three different steps:

- Loading optical absorption spectra of inks.
- Setting-up of model parameters (concentration of the inks, component replacement strategy, parameters of Saunderson correction, absorption spectrum of paper used, resolution...).
- Computation of the gamut surface and its volume, 3D visualization and saving.

The main window (cf figure 4) controls to the settings of the model parameters. The button "Gamut Generator" allows us to execute the C/C++ program for the Color Generator but also the surface reconstruction. Finally, the window opened by the button "3D visualization" shows the result using an interactive-openGl window controlled by Vtk.⁷ The user can save this image in a .png format as a screenshot but also the Gamut data obtained in a VRML or Vtk format.

Figures 3 and 4 represent the 3D visualization of the color gamut for a hypothetical inks set (2°, D50) for two model settings.

Figure 3 represents the Gamut based on Gray Component Removal and without Saunderson correction. This refers to a "transmission" case and a comparably small Gamut volume of 34119 unit colors is obtained.

Figure 4 represents the same ink set, ink concentrations and GCR settings, but with the Saunderson correction activated. The Gamut volume corresponds to a "reflection" case is considerably larger than before: 149773 unit colors.

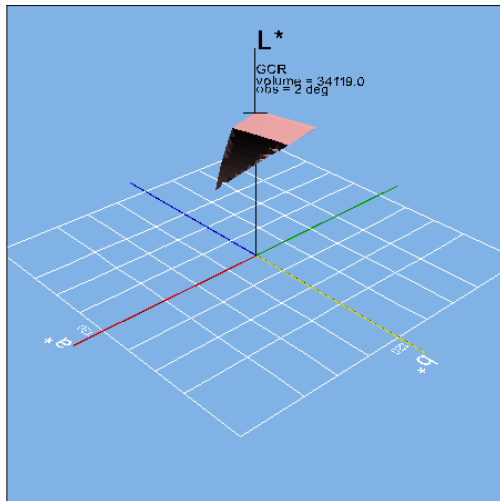


Figure 3. Example of a Gamut of hypothetical ink set based on Gray Component Removal and without Saunderson correction ("transmission case")

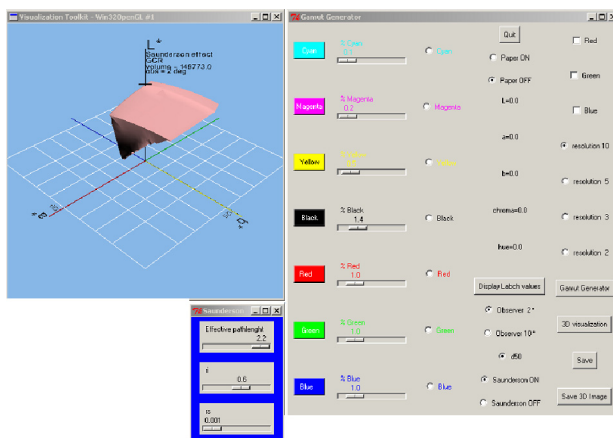


Figure 4. Same ink set as in Fig. 3 but with Saunderson correction ("reflection case")

Conclusion

A Color Generator and visualization tool for up to seven inks is presented, which – based on the assumption of an ideal printing process – allows the calculation of the potential color gamut of a given ink set.

References

1. US 6,633,408 B1 spectral modelling of photographic printing based on dye concentration
2. Marc Mahy, Gamut calculation of color reproduction devices, 4th IS&T/SID Color Imaging Conference, Scottsdale, Arizona, Volume 4, p 145-150, (1995)
3. Ryoichi Saito and Hiroaki Kotera, Extraction of Image Gamut Surface and Calculation of its Volume, 8th IS&T/SID Color Imaging Conference, Scottsdale, Arizona, Volume 4, p 330-334, (2000)
4. Gustav J. Braun and Mark D. Fairchild. Techniques for Gamut Surface Definition and Visualization, IS&T/SID Fifth Color Imaging Conference, pp.147-152, (1997)
5. Patrick Emmel. Modèles de prédiction couleur appliqués à l'impression jet d'encre. Phd: Ecole polytechnique fédérale de Lausanne, (1998)
6. John K. OUSTERHOUT. *Tcl and The Tk Toolkit*. Addison wesley éditions, juillet 1995. Addison Wesley professional computing series.
7. William J. Schroeder, Kenneth M. Martin and Bill Lorenson. *The visualization toolkit an object-oriented approach to 3D graphics*. Kitware, Inc., second edition, (2001)
8. N. Amenta, S. Choi, Tamal K. Dey and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *Intl. J. Comput. Geom. Appl.*, Vol. 12, 125-141, (2002)
9. Tamal K. Dey and Samrat Goswami. Tight Cocone: A water tight surface reconstructor. *Proc. 8th ACM Sympos. Solid Modeling Appl.*, 127-134, (2003)

Biography

Cédric Marchessoux received his PhD in Electronics Engineering at the SIC laboratory (signal and image processing and computer sciences) at the University of Poitiers/France in 2003. He has followed studies of theoretical Physics applied to medical imaging and is currently working as a post doctorate Marie Curie industrial fellow at the material research laboratories of Agfa-Gevaert, Mortsel under supervision of Dr. Jürgen Jung.