Techniques for Gamut Surface Definition and Visualization

Gustav J. Braun and Mark D. Fairchild
Munsell Color Science Laboratory
Rochester Institute of Technology
Rochester, New York

Abstract

Color gamut mapping plays a crucial role in color management. Accurate techniques for defining and visualizing a device's gamut boundary are very important in the design of robust color gamut mapping algorithms. A novel technique for defining the surface of a color imaging device's gamut in CIELAB $L^*C_{ab}^*h_{ab}$ color space using a triangulation and interpolation process is presented. This process provides an accurate approach for gamut surface fitting, from measured or modeled data, that is independent of gamut concavity or convexity. The results of a goodness-of-fit test indicate that the gamut surface can be predicted to a mean ΔE_{ab}^* of 1.1, for the CRT gamut tested. In addition the $L^*C_{ab}^*h_{ab}$ space has been shown to be useful for several gamut mapping and visualization tasks. Finally, an $L^*C_{ab}^*h_{ab}$ gamut is given for an HP Deskjet 870Cxi inkjet printer that was derived from measured data.

Introduction

Color gamut mapping is an integral part of color management. It is important to be able to accurately model the gamut surfaces for all of the devices used in a color image reproduction chain. Often, the source and destination gamuts in this chain are dissimilar. Therefore, in order to obtain high quality color reproductions between these devices, some type of color gamut mapping must occur. In order to perform gamut mapping in a visually effective manner a description of a device's color gamut needs to be obtained. [1,2,3,4] Thus far, little has been published regarding the specification of a color imaging device gamut boundaries directly from measured data.

This research gives a technique to define the gamut boundaries of color imaging devices, and modes of display that are useful for visualizing the gamut volume in two and three dimensions. The gamut surface estimation is based on well-established surface-fitting procedures that are commonly used by mathematical analysis software packages such as MATLAB™ and IDL™ for generating a uniform grid of points based on a set of non-uniformly space input points. The approach presented here applies these relatively straight-forward processes to simplify the estimation of an imaging devices color gamut.

The following sections will provide: 1.) a description of a triangulation and interpolation process used to convert non-uniformly spaced color data, derived from a device RGB cube, into a CIELAB $L^*C_{ab}^*h_{ab}$ ("mountain-range") representation of the color gamut; 2.) a "goodness-of-fit" test that was used to gauge the accuracy of the gamut surface estimation process; 3.) the benefits of using the "mountain-range" gamut for gamut-mapping and gamut-visualization; 4.) an example "mountain-range" gamut for an HP Deskjet 870Cxi inkjet printer that was derived from spectrophotometrically measured data.

CIELAB $L^*C_{ab}^*h_{ab}$ Gamut Specification

In order to perform color gamut mapping, image data need to be in a reference color space that is visually based. This requires transforming device-dependent image data and device color gamuts into a reference visual color space. The device dependent representation of a color imaging device's gamut can be generalized by its RGB cube. The surface of the RGB cube can be thought of as all one and two primary mixtures as well as all 3 color mixtures when at least one of the primaries is at a maximum. The eight corners of this cube correspond to device red, green, blue, cyan, magenta, yellow, black, and white. The device-independent (e.g., CIE XYZ or CIELAB) or viewing-conditions-independent (e.g., RLAB[5]) representation of a color imaging device's gamut is usually a non-linear transformation of its device dependent-representation. Calculation of the gamut surface for a color imaging device consists of transforming the RGB cube surface into corresponding values in a reference color space, either using a physical model or printing and spectrophotometrically measuring selected values from the surface of the RGB cube after appropriate transformation to

CMYK if necessary.

Once the gamut surface data are in the desired reference color space, such as CIELAB, computer graphics algorithms such as convex hull routines can be used to form a polygon mesh encompassing the data.[4] This process works well if the gamut surface, in the reference color space, is convex. However, if the surface has concavities they will be masked by the polygon mesh used to define the gamut boundary. Color gamut-mapping algorithms may use these polygon mesh structures and "ray-tracing" or other computer graphics procedures to map out-of-gamut points into the color space regions encompassed by the destination gamut.

Some gamut mapping and visualization tasks are better or more efficiently performed when a device's gamut is represented in cylindrical coordinates such as CIELAB $L^*C_{ab}^*h_{ab}$. In this form, the C_{ab}^* component on the gamut surface is functionally related to L^* and h_{ab}. That is, for any hue angle, each lightness is represented by a single chroma value. In the CIELAB representation of a gamut, L^* is not functionally represented in the a^*b^* plane since each $[a^*,b^*]$ point will have two associated L^* values. For visualization purposes the $L^*C_{ab}^*h_{ab}$ representation makes it very easy to view the structure of the gamut.

Points that are uniformly spaced on a device RGB cube will typically be non-uniformly spaced in CIELAB due to the non-linear relationship between the two spaces. When these data are converted into $L^*C_{ab}^*h_{ab}$ and projected into the L^*h_{ab} plane they lie on an irregular grid. In order to convert this irregular or non-uniform set of points into a regular grid, interpolation is needed. One such interpolation process involves using triangular interpolation between the data points in the L^*h_{ab} plane.

Triangulation and Interpolation

The current technique transforms the gamut surface data into cylindrical CIELAB coordinates (e.g., $L^*C_{ab}^*h_{ab}$) and performs a triangulation and gridding process to determine the surface of the gamut. The triangulation of the data is performed by projecting the nonlinearly spaced $L^*C_{ab}^*h_{ab}$ data from the device RGB cube onto the L*h_{ab} plane. The data in the L^*h_{ab} plane are grouped into triangles using the inherent "connectivity" associated with the points from the RGB cube, Figure 1. The vertices of this mesh are measured or modeled data from the surface of the RGB cube. Using the triangle list and the corresponding C_{ab}^* for each triangle vertex, a uniform L^*h_{ab} grid of C_{ab}^* values is interpolated using triangular linear interpolation, Figure 2. In order to assure that the chroma at $h_{ab}=0°$ and $h_{ab}=360°$ is the same, the original data are treated as periodic over h_{ab} and replicated on either side of the original data, Figure 3.
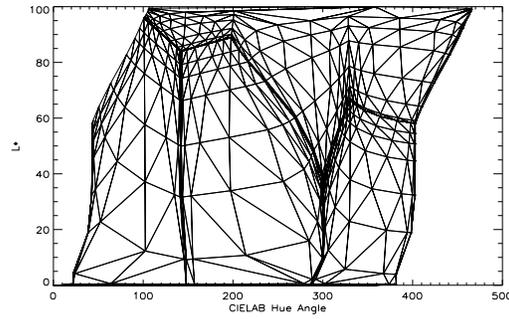
Figure 1: Projection and triangulation of uniform RGB cube vertices into L^*h_{ab} plane.

$$u=Au/A, \quad v=Av/A,$$
$$w=Aw/A$$

$$C^*(u,v,w) = uC_u^* + vC_v^* + wC_w^*$$
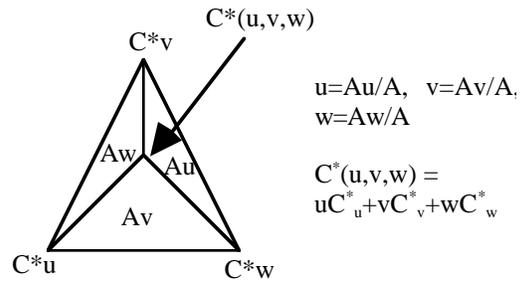
Figure 2: Illustration of triangular linear interpolation.
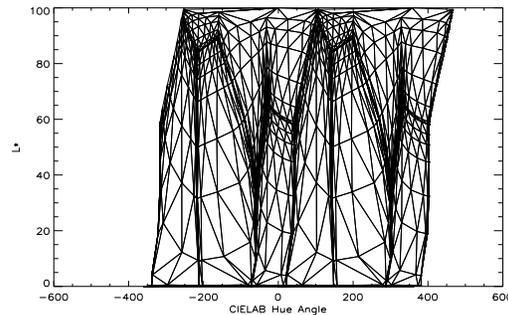
Figure 3: Periodic replication of the measured RGB cube data to insure connectivity between $h_{ab}=0°$ and $h_{ab}=360°$.

The uniform grid points are interpolated for steps of $\Delta h_{ab} = 1°$ and for $\Delta L^* = 1$. The resulting uniform C_{ab}^* grid in the L^*h_{ab} plane is represented by a 101x361 element matrix. The chroma, C_{ab}^*, of the gamut surface for any $[L^*, h_{ab}]$ coordinate can be estimated from the uniform C_{ab}^* grid using bilinear or cubic-convolution interpolation procedures. The following series of figures gives different representations of the uniform grid of C_{ab}^* values for a typical CRT device with no external or internal flare terms present (i.e., assume the device black can achieve an $L^*=0$). In Figure 4, the intensity of each pixel is proportional to the chroma (i.e., low/high intensity corresponds to low/high chroma). The surface plot shown in Figure 5 has been referred to as a "mountain-range" gamut based on the peak-

like structure of the gamut surface in cylindrical CIELAB coordinates. The height of the surface from the L^*h_{ab} plane is given by C_{ab}^*.

The main assumptions made in this process are that the chroma of the "black-point" and the "white-point" drops off to zero. This is necessary since in the $L^*C_{ab}^*h_{ab}$ representation the "black-point" and the "white-point" are replicated for all hue angles at the L^* for which they occur. Therefore, the C_{ab}^* for these two points are forced to zero if they are not already zero. In general this assumption has held true for the devices examined thus far.
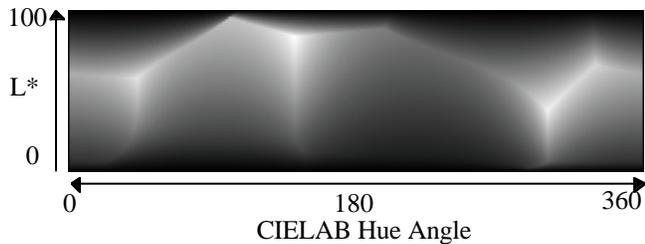
Figure 4: $L^*C_{ab}^*h_{ab}$ representation of a CRT gamut generated using triangulation and interpolation. Vertical scale represents L^* and horizontal scale represents $h_{ab}=[0°,360°]$.
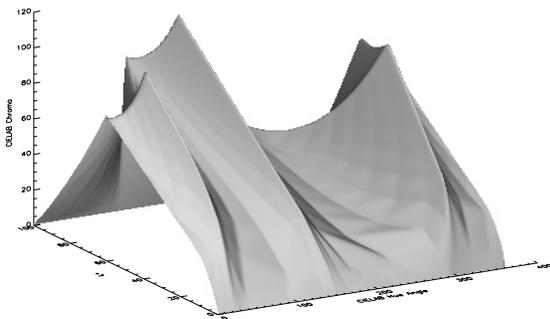
Figure 5: "Mountain-range" representation of the $L^*C_{ab}^*h_{ab}$ CRT gamut generated from triangulation and interpolation.

A very useful feature that results from storing the gamut surface data in the $L^*C_{ab}^*h_{ab}$ representation is that individual hue angle slices can be quickly extracted from the C_{ab}^* matrix and visualized by a simple 2D plot, Figure 6. These slices prove to be very useful in designing color gamut mapping algorithms that are customized on a hue-angle dependent basis. If the gamut data were represented as a 3D wire-frame mesh in CIELAB space, some type of estimation process would be required to extract a slice profile of the gamut surface for a give hue angle. This might involve a series of "ray-tracing" steps where the gamut intersection points would be located for a series of L^* values for the given hue angle. Such a process is more computationally demanding than looking-up or interpolating values from a 2D matrix.
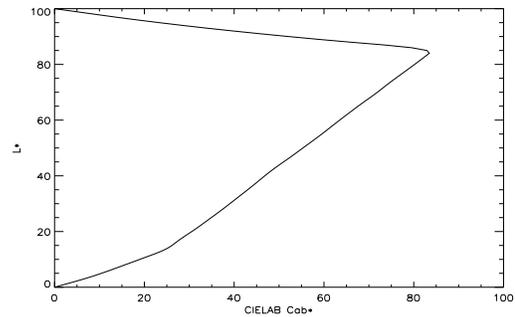
Figure 6: Illustration of a "slice" taken from the "mountain-range" representation of CRT gamut.

It may also be desirable to transform the uniform $L^*C_{ab}^*h_{ab}$ grid into rectangular coordinates, $L^*a^*b^*$, Figure 7. This will generate a highly faceted wire-frame model of the triangulated and interpolated gamut. Neighboring points in the $L^*C_{ab}^*h_{ab}$ representation remain neighboring points in CIELAB. Therefore, the connectivity of a polygon mesh in $L^*C_{ab}^*h_{ab}$ space, prior to the coordinate transformation, is the same in CIELAB representation. As such, no "hulling" procedures are required to produce a polygon mesh in CIELAB. This polygon mesh can then be used for gamut mapping, for gamut mismatch visualization, and for visualization of image pixel data within the source and destination gamuts. All of these processes aid in the development of color gamut mapping algorithms.
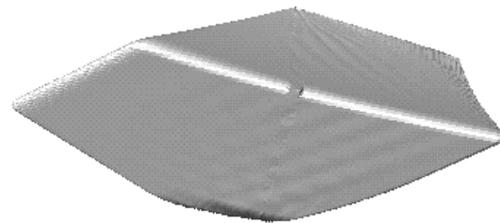
Figure 7: CIELAB "wire-frame" representation of the triangulated and interpolated CRT gamut.

Noise Filtering and Smoothing

This process of gamut surface estimation offers a straight forward process to reduce the effects of measurement noise and print-to-print variability that exists in measured RGB cube data used to generate the gamut surface. The "mountain-range" gamut that results from the triangulation and interpolation process is simply a matrix. If there is noise in the data that was used to generate the "mountain-range", the gamut surface will not be smooth, Figure 8. By convolving the "mountain-range" gamut with an averaging

kernel the noise can be reduced, Figure 9. Other filter functions could be designed to help reduce the noise more optimally if information were available regarding the nature of the noise structure. Minimally, operations such as median filtering can be applied for this purpose.
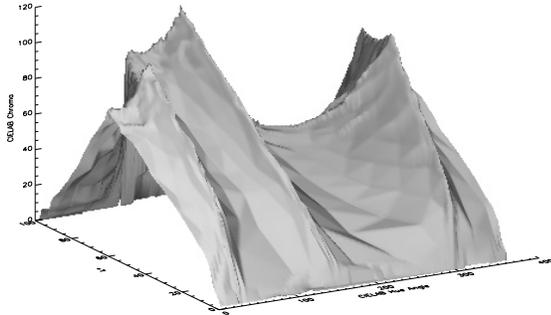
Figure 8: "Mountain-range" gamut of a CRT with simulated measurement and print-to-print variability ("white") noise.
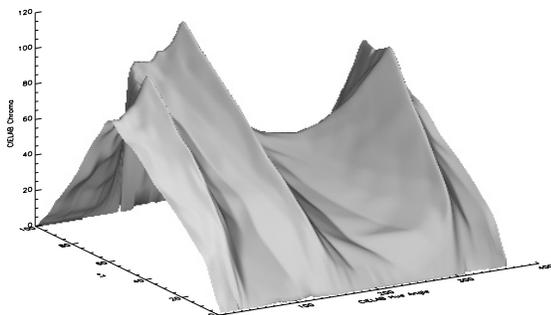
Figure 9: Noise Filtered "Mountain-range" gamut of the gamut shown in Figure 8.

Another useful feature of the "mountain-range" representation of a gamut is that the gamut data can be pre-filtered to reduce the effects of spatial aliasing in the gridding and interpolation process. For example, in order to control the effects of aliasing that result from the gridding process, a "mountain-range" gamut can be over-sampled at a rate of $\Delta h_{ab}=0.1°$ and $\Delta L^*=0.1$. The resulting uniform grid is then convolved with an anti-aliasing prefilter such that when the matrix is resampled onto a grid with spacings of $\Delta h_{ab}=1°$ and $\Delta L^*=1$, aliasing does not occur.

Error Analysis

The objective of this test was to determine the accuracy of the "mountain-range" gamut produced for a uniform grid of C_{ab}^* values compared to those values predicted by a gamma-gain-offset (GOG) CRT model[7].

Procedure

The CIELAB data for the CRT gamut were generated using a GOG model with no flare terms present. The CRT "mountain-range" gamut was generated using the following procedure:

1. Generate RGB node points that made up a 20x20 grid of non-linearly spaced points for each of the six faces of the RGB cube. The spacing of the grid points on the faces of the RGB cube were most heavily weighted toward the edges and corners of the cube.
2. Convert the RGB digital counts into CIELAB using the GOG model for the CRT.
3. Convert CIELAB data to CIELAB $L^*C_{ab}^*h_{ab}$.
4. Generate a uniform, regular grid (in the [L*,h_{ab}] plane) of C_{ab}^* values using the triangular interpolation process. The grid spacing was $\Delta h_{ab}=1°$ and $\Delta L^*=1$. A 3x3 "Boxcar" averaging filter was applied to reduce the effects of aliasing. In the convolution, points at the edges of the C_{ab}^* image were left "as-is" to avoid wrap around effects of circular convolution.

Once the "mountain-range" gamut was defined, a random sampling of 6000 *surface points* for an RGB cube were generated. These points were converted to CIELAB $L^*C_{ab}^*h_{ab}$ values using the GOG model. For each of the 6000 points a chroma value was interpolated from the C_{ab}^* "mountain-range" gamut at the [L*,h_{ab}] values corresponding to the modeled value. An error analysis was then performed that compared the *modeled* CIELAB values (from the GOG model) to the *estimated* CIELAB values derived from the mountain-range (Note: The difference between *modeled* CIELAB and *estimated* CIELAB is that the chroma value for the *estimated* CIELAB were derived from the "mountain-range" gamut).

Results and Discussion

Table 1, shows that there is little difference between the modeled-gamut predicted from the GOG device model for the CRT and the "mountain-range" estimated gamut. A plot of the histogram of ΔE_{ab}^* errors for all 6000 points confirms that the fit between the estimated "mountain-range" gamut and the device model are accurate, Figure 10.

Table 1: Results of error analysis. Comparison of "mountain-range" predicted gamut surface to gamut surface predicted by CRT GOG model

Number of Points used in Analysis	6000
Mean ΔE_{ab}^*	1.06
Variance in ΔE_{ab}^*	3.74
Max ΔE_{ab}^*	30.15
No. of Points w/$\Delta E_{ab}^* > 5.0$	126
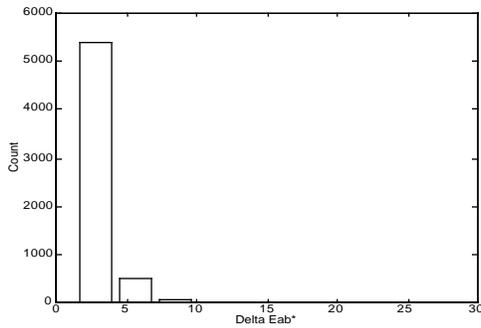
Figure 10: Histogram of ΔE_{ab}^{*} errors between GOG modeled points and points estimated from the "mountain-range" gamut.

There are, however, a few points about the results that should be discussed. The first is the relatively large *maximum* ΔE_{ab}^{*} that resulted from the analysis. The graphs given in Figure 11 represent vector error plots for all of the test points that resulted in ΔE_{ab}^{*} values *greater* than 5.0 (126 points out of 6000). The first plot in Figure 11 shows that the majority of the errors are in estimated chroma (i.e., the *estimated* chroma is lower than the *modeled* chroma). Nearly all of these errors occur in about the same hue angle region (i.e., approximately between 90 degrees and 130 degrees, h_{ab}). These hue angles line up well with the yellow CRT secondary. This plot shows that the majority of large errors are in the estimation of b^{*}. The $[C_{ab}^{*}, L^{*}]$ plot indicates that the large errors occur for the high lightness samples (e.g., $L^{*} > 80$).

The reason for these large errors is that in forming the C_{ab}^{*} grid ("mountain-range"), the location of the yellow primary gets shifted *slightly* in hue angle as a result of the discrete location of the grid points. Since the peak for the yellow primary is so steep coming up from "white" ($L^{*} = 100$), any mis-alignment in the uniform interpolation grid may cause large errors in the estimated chroma. Similar errors may occur at other peak locations. Their effect would not be as pronounced as that for the yellow primary since the data at these peaks do not vary as rapidly as the data near the yellow peak. (The yellow peak occurs over a ΔL^{*} region of 6 units and a Δh_{ab} region of ~2°). Based on the grid spacing the shift would be on the order of 0.5° in h_{ab} and up to 0.5 units of L^{*}. Since the C_{ab}^{*} value of the "yellow peak" does not appear to be altered, just its location, this phenomenon should not effect the results of any gamut mapping experiments or the appearance of gamut mapped images.
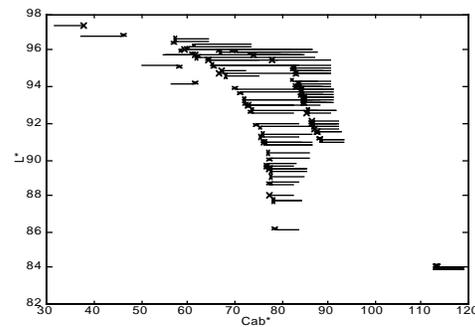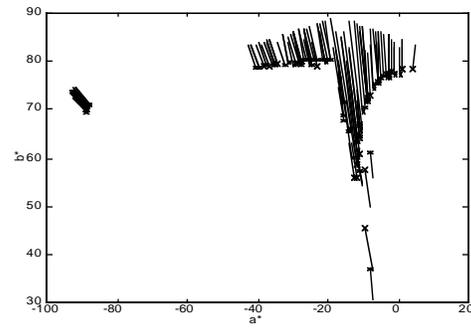
Figure 11: CIELAB error vector plots for sample points whose ΔE_{ab}^{*} was greater than 5.0 ("o" represent modeled values and "x" represent "mountain-range" estimated values).

Advantages of "Mountain-range" Gamuts for Gamut Mapping

Gamut Specification:

The gridding and interpolation process in the $L^{*}C_{ab}^{*}h_{ab}$ color space has several distinct advantages when compared to specification in CIELAB using convex hull algorithms. First, data concavity is not an issue. The surface fitting process will work equally well on concave data sets as convex data sets. Additionally, it is easy to determine how interpolated surface points are connected to other points in the lattice based on their position in the $L^{*}h_{ab}$ plane.

Device models are not necessary to generate the gamut surface of a device; the gamut can be estimated from measured data directly. This process proves to be a robust surface fitting algorithm in the presence of measurement noise and/or print-to-print variability. Measured data is inherently noisy. This noise will cause micro-concavities in the surface structure of the gamut. Convex hull routines will always over-predict the surface of the gamut by masking the micro-concavities. This process will fit this surface texture automatically. Also, the ability exists to filter out unwanted noise through straight forward image processing.

Gamut Mismatch Estimation:

Determination of the gamut difference between two devices whose gamuts are represented as "mountain-ranges" requires simply subtracting the two "mountain-ranges". The resulting matrix will contain both the direction (+/-) and the magnitude of the gamut mis-match is in C_{ab}^{*} units. This makes it easy to determine the regions in which the source gamut is in-gamut or out-of-gamut of the destination gamut. This type of information is key to performing gamut mapping.

The benefits of this process extend to evaluation of individual image pixels as well as the source and destination gamuts. To determine if a given source image pixel is in or out of gamut of the destination gamut, the C_{ab}^{*} value of the destination gamut surface is estimated by interpolation for the $[L^{*}, h_{ab}]$ coordinate of the source pixel. The C_{ab}^{*} value of the source pixel is then compared to that of the destination gamut. If this difference is positive the source pixel is out-of-gamut of the destination gamut. If the difference is negative the source pixel is within the gamut of the destination device. These types of processes using a CIELAB wire-frame representation of the gamut require more computationally complex calculations and "ray-tracing" algorithms.

Gamut Visualization:

As was presented above, visualization of the gamut surface structure is easy using the "mountain-range" gamut representation. Location of the hue angles of the device primaries is easily performed. It is straightforward to extract slices of the gamut for individual hue angles. The "mountain-range" representation is also useful for the design of gamut mapping routines. Many algorithms perform the gamut mapping, inherently, in the $L^{*}C_{ab}^{*}h_{ab}$ space (e.g., ones that preserve hue angle). As such it is easy to visualize how the data need to be processed using the "mountain-range" representation.

Gamut Mapping:

Several types of gamut mapping algorithms can be processed very efficiently using the $L^{*}C_{ab}^{*}h_{ab}$ gamut representations. These algorithms include chroma clipping while maintaining L^{*} and h_{ab}, minimum ΔE_{ab}^{*} clipping to the surface of the gamut using orthogonal projection and preservation of CIELAB hue angle, along with other lightness and chroma scaling techniques. Essentially, representing the gamut in $L^{*}C_{ab}^{*}h_{ab}$ simplifies the location of a point within a gamut and the location of the point towards which the reference point is mapped.

Example CIELAB $L^{*}C_{ab}^{*}h_{ab}$ Gamut for an HP Deskjet 870Cxi Inkjet Printer

The "mountain-range" gamut shown in Figure 12 represents the gamut for an HP Deskjet 870Cxi inkjet printer using glossy paper. The grid spacing for the gamut is in steps of $\Delta h_{ab}=1°$ and $\Delta L^{*}=1$. The data used in the gridding process came from an RGB cube with 20x20 points per face. A 5x5 "Boxcar" averaging filter was applied to reduce the noise in the data. The L^{*} axis in Figure 12 is scaled between 0 and 100. These are normalized coordinates. The L^{*} range for this printer is {12,100}.
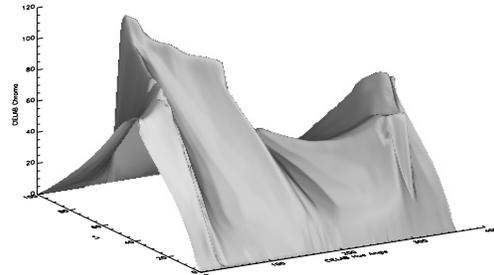
Figure 12: Sample "mountain-range" gamut generated from spectrophotometrically measured CIELAB data.

Conclusions

The results of the research and the experiments presented above indicate that a CIELAB $L^{*}C_{ab}^{*}h_{ab}$ gamut can be generated that will accurately model the surface structure of a color imaging devices gamut. This gamut fitting process allows for the generation of an imaging device's color gamut without the necessity of a device model. The triangulation and interpolation process presented will operate regardless of whether the source data is convex or concave in nature. The ability to filter out measurement noise and sample variability is a powerful feature of this technique.

References

1. McBride, T.O., "Analysis of Digital Printer System Color Gamut", IS&T NIP12: International Conference on Digital Printing Technologies, pp 117-120, 1996.
2. Herzog, P.G., "Analytical Color Gamut Representations", Journal of Imaging Science and Technology, Vol. 40, Num. 6, pp 516-521, 1996.
3. Inui, M., "Fast Algorithm for Computing Color Gamuts", Color Research and Applications, Vol. 18, Num. 5, pp 341-348,1993.
4. Balasubramanian, R. and Dalal, E., "A Method for Quantifying the Color Gamut of an Output Device", Final Program and Abstracts of IS&T/SPIE's 9th Annual Symposium, Electronic Imaging: Science and Technology, pp 114, Feb. 9 - 14, 1997, San Jose, CA.
5. Fairchild, M.D., "Refinement of the RLAB Color Space", Color Research and Application, Vol. 21, pp 338-346, 1996.
6. Preparata, F.P. and Shamos, M.I., "*Computational Geometry - an Introduction*", Springer-Verlag, 1985.
7. CIE Technical Report 22, "The Relationship Between Digital and Colorimetric Data for Computer Controlled CRT Displays", 1996.