Journal of Imaging Science and Technology® 54(5): 050502–050502-11, 2010.
© Society for Imaging Science and technology 2010

Evaluation of Algorithms for the Determination of Color Gamut Boundaries[1]

Arne M. Bakke, Ivar Farup▲ and Jon Y. Hardeberg▲
Gjøvik University College, Gjøvik, Norway
E-mail: jon.hardeberg@hig.no

Abstract. *Several techniques for the computation of gamut boundaries have been presented in the past. In this article we take an in-depth look at some of the gamut boundary descriptors used when performing today's gamut mapping algorithms. We present a method for evaluating the mismatch introduced when using a descriptor to approximate the boundary of a device gamut. First, a visually verified reference gamut boundary is created by triangulating the gamut surface using a device profile or a device characterization model. The different gamut boundary descriptor techniques are then used to construct gamut boundaries based on several sets of simulated measurement data from the device. These boundaries are then compared against the reference gamut by utilizing a novel voxel based approach. Results from experiments using several gamut boundary descriptors are presented and analyzed statistically. The modified convex hull algorithm proposed by Balasubramian and Dalal performs well for all the different data sets. © 2010 Society for Imaging Science and technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2010.54.5.050502]

INTRODUCTION

The construction of a gamut boundary descriptor (GBD) is the first step in the process of performing gamut mapping. While there has been extensive[1] research done on the performance of gamut mapping algorithms (GMAs), little has been done to compare the performance and validity of the commonly used GBDs. Since many of the GMAs depend on finding the intersection between lines and a gamut boundary, any inaccuracies introduced by using a GBD that fails to accurately represent the gamut boundary result in errors later in the gamut mapping process. If the GBD overestimates the gamut volume in some areas of the color space, the GMA may result in colors that are still not reproducible on the output device. Similarly, gamut underestimation leads to unnecessary image gamut compression, leaving parts of the destination gamut unused. GMAs that utilize both the source and the destination gamut may further magnify the problem, since both gamut boundaries may contain errors.

Although exact determination of the gamut boundary is important to the result of any GMA, different GBDs have been used seemingly arbitrarily by different researchers. Similarly, for gamut metrics like, e.g., the gamut volume, the method used for determining the GBD is often left unmentioned and the results are thus hardly comparable. No systematic comparison of existing methods for determining the GBD is known to the authors.

The purpose of the present work is to provide a thorough comparison of the most commonly used methods for determining the GBD. The methods are first described and discussed in terms of algorithmic complexity, feasibility of the resulting geometric structure and specific implementation issues. Then, the algorithms are compared with respect to how accurately they are able to reproduce the surface of known gamuts from different data sets extracted from the gamuts and for different choices of algorithm parameters. It turns out that there is great variability of the performance of the algorithms, particularly for the most sparse data sets.

BACKGROUND AND STATE-OF-THE-ART

There are several known approaches for determining the GBD. Some depend on knowledge concerning the characteristics of a device, and are therefore only applicable to device gamuts of the specific device type. These methods construct a device model, and the gamut boundary follows from physical limits of the device, e.g., ink coverage. Approaches based on analytical models are clearly unsuitable for color sets that do not follow such constraints, while other methods for gamut boundary computation may be used also to determine the gamut boundary of images. Such methods usually require measured data sets as input, in the form of colors (points) in a color space.

Model based methods

MacAdam[2] presented an early attempt at a model based gamut. An approximation of the gamut was found by assuming box-shaped colorant reflectance and calculating CIEXYZ tristimulus values. The gamut of a printing system can also be determined by using the Kubelka-Munk[3] equations, as shown by Meyer et al.[4,5] in 1993. Mahy used the Neugebauer[6] equations to calculate the gamut of a multi-ink printing system.[7]

Inui[8] introduced an algorithm for the computation of printer based color gamuts, using an assumed correspondence between color space and dye amount space. Herzog introduced an analytical mathematical description of color gamuts called gamulyts,[9,10] where the gamut is represented

▲IS&T Member.
[1]A preliminary version of this work was presented at IS&T/SID's 14th Color Imaging Conference, November, 2006, Scottsdale, AZ.

Received Apr. 26, 2010; accepted for publication May 20, 2010; published online Sep. 9, 2010.
1062-3701/2010/54(5)/050502/11/$20.00.

by a deformed cube. By using a set of distortion functions, a cube is deformed to fit the color gamut. An extension of this model can also be applied to systems with more than three colorants.

Point based methods

It is possible to obtain a device gamut easily by assuming that the gamut boundary of a device is preserved between device dependent and device independent color spaces. A simple approximation of the gamut can be found by measuring the colors that make up the extreme points of the gamut. Stone et al.[11] proposed that the gamut can be represented by planes connecting these extreme points.

In order to represent the gamut more accurately, Bolte[12] performed a direct triangulation of measured colors found by printing patches that make up a regular structure in the device color space. One of the problems with this approach is that characteristics of the printing process and the color space transformation can lead to the order of the tetrahedra vertices being reversed, and internal points in the device color space structure may thus end up outside the triangles that form the surface of the regular structure when converted to CIELAB. This method can be further improved by enforcing a check for mirrored tetrahedra in the device independent color space, and testing for points on the outside of the elements that make up the surface of the regular structure.[13,14]

If the data measurements do not follow any structure, it is still possible to determine the gamut boundary by applying one of several geometric algorithms to find the surface of the points. The convex hull of the measurement data can be found by using, e.g., the quickhull[15] algorithm. This results in a convex approximation of the gamut,[4,5] and has been used to find gamuts from ICC profiles.[16] The main problem with this approach is that device gamuts usually have some concave sides. If the data originates from a reasonably well-behaved printer, it is possible to compute the convex hull in a linearized dye density space,[17] where the points are assumed to be more convex.

Balasubramian and Dalal[18] presented an improvement of the standard convex hull method. By introducing a nonlinear method as a preprocessing step before the convex hull is found, the gamut surface is given the ability to follow concavities in the original data set.

A different approach to gamut determination is to find maximum chroma for cells having a specific lightness and hue. By imposing a regular grid structure on these points, triangulation can result in a gamut surface,[19,20] commonly referred to as a "mountain range." Segment maxima[21] is a related method that performs a subdivision of the color space into segments based on the polar coordinates of the colors. The segment maxima technique has also been applied to image data,[22,23] where the mass center can be used as the origin of the spherical coordinate system.

Cholewo and Love[24] used alpha shapes[25] to find the gamut boundary of both devices and images. The alpha parameter controls the level of details of the computed shape, and Cholewo and Love suggest that the optimal α should be found by interactively changing the shape.

Giesen et al.[26] proposed the use of a discrete flow complex to compute image gamuts. By using a grid representing the relevant part of the color space, and computing the distance to the nearest sample in the point data, a discrete three-dimensional (3D) map of the color space can be found. By comparing the grid value at a certain grid position to its neighbors, a flow is established. Traversal of this structure then decides which grid points are considered part of the gamut.

THE ALGORITHMS

While many GBDs have been suggested for use in, e.g., color gamut mapping, few are in use today. We will study the performance of some commonly used GBDs, as well as some that have been shown to have advantages when compared with these algorithms. We limit our investigation to algorithms that can be applied to any type of generic color data in a three-dimensional opponent color space, thereby excluding model-based algorithms. The convex hull is used by many when computation of gamut volumes is desired, although the algorithm is known to overestimate the gamut volume. Bala's modified convex hull represents a great improvement over the standard convex hull, but is not as well known. The segment maxima GBD is found in the reference implementation of gamut mapping algorithms distributed by the CIE. The alpha shapes algorithm is utilized in many fields for finding the shape of a set of points, but is not commonly used to find color gamuts. The alpha parameter needs to be decided, and there is no standard method for doing this. We will also investigate further the performance of the uniform segment visualization method suggested by Bakke et al.[29] that has recently been proposed as an alternative to the segment maxima algorithm.

We have implemented the different methods for constructing the gamut boundaries using the ICC3D[27] application as a basic framework for the comparisons. This tool can visualize a variety of GBDs, allowing visual verification of our results. Its modular architecture is particularly suited for implementation of new algorithms.

Convex hull

The convex hull of a set of points X is the smallest convex set containing X. Any point in space that can be defined as a convex linear combination of the points defining the convex set, i.e., a linear combination with weights greater than or equal to 0 and sum equal to 1, is a part of the convex hull of the data set.

The convex hull of a set of colors is often used as an approximation of a color gamut. The convex hull can be found using, e.g., the quickhull algorithm. The availability of this algorithm in tools commonly used for data analysis is likely to be one of the main factors advocating its use. The algorithm has also been in use for a relatively long time, an its properties are well known. The expected complexity of the algorithm is $O[N \log(N)]$, but it has a maximum limit of $O(N^2)$ The convex hull is guaranteed to contain all of the

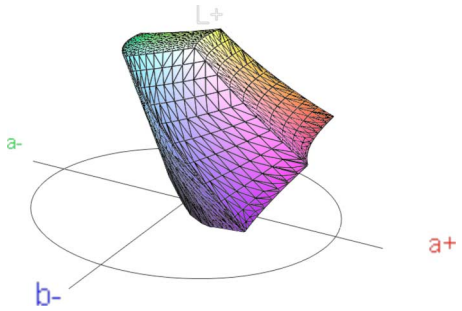

Figure 1. A gamut surface found by using the GBD proposed by Balasubramian and Dalal.

data points, which is an important advantage when comparing it to other GBDs.

However, the convex hull overestimates the volume of gamuts. Color gamuts in perceptual color spaces generally have concave surfaces that the convex hull algorithm does not detect.

Modified convex hull

The modified convex hull algorithm is based on the improvements suggested by Balasubramian and Dalal,[18] resulting in surfaces similar to the one illustrated in Figure 1. Before the convex hull is computed, new vertex coordinates $\vec{p}'$ are calculated from the original position $\vec{p}$ using a gamma function based on the distance to the color space center $\vec{c}$, and a parameter, γ.

$$\vec{p}' = |\vec{p} - \vec{c}|^{\gamma} \frac{\vec{p} - \vec{c}}{|\vec{p} - \vec{c}|} + \vec{c}. \tag{1}$$

By using a γ between 0 and 1, the data are made more convex before the convex hull algorithm is applied. A γ value of 1 does not alter the data, and is equivalent to a standard convex hull, while values closer to 0 move all colors closer to the surface of a sphere. By increasing the convexity of the data, points close to the convex surface are made part of the surface. The result is a gamut boundary that more closely follows the perceived surface of the data set, including concavities. However, if the data set contains internal points, the choice of a smaller γ increases the probability that these points are added to the surface.

When applying the modified convex hull algorithm to a data set consisting of surface colors in order to construct a surface composed of polygons, it might seem obvious that using a very small γ is the best solution, since this ensures that all the points are made part of the surface. However, there are two important considerations that make such an assumption false:

(i) The surface may fold in on itself since the curvature of the gamut surface can cause lines between the chosen center point and the gamut surface to cross the gamut boundary twice before the final intersection with the gamut surface. This creates the possibility that points are made part of the surface structure in the wrong place, making the gamut surface appear jagged with artificial holes.

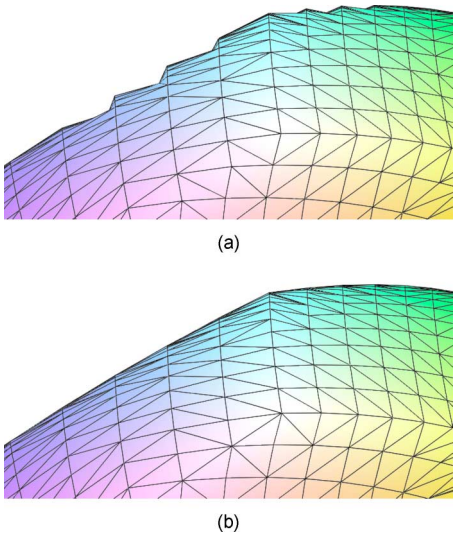

Figure 2. The effect of the γ parameter on the construction of gamuts using the modified convex hull algorithm. (a)The use of a small γ value in the pre-processing step has caused artifacts in the surface. (b) A slightly larger γ results in a surface without artifacts.

(ii) While a very small γ value ensures that all of the surface points are part of the resulting surface, the operation may lead to the inclusion of a set of erroneous edges that replace edges that should be part of the surface. This can typically be seen most clearly along edges between the primary and secondary color corners of the gamut, where the γ expansion can lead to notches in the gamut surface. Figures 2(a) and 2(b) illustrate this effect.

Convex hull in CIEXYZ

Device gamut measurements generally do not constitute completely convex objects in CIELAB or related color spaces. Guyler[28] proposed that the convex hull algorithm be performed in CIEXYZ, and the resulting surface transformed to the desired color space. Guyler argued that color data tend to be more convex in the CIEXYZ space, and that this approach thus results in a better approximation.

First, the data points are converted from their original color space to CIEXYZ. The convex hull of these vertex coordinates is computed and represented by a list of facets, and the connectivity and vertex information of these facets is used to construct a surface in the original color space by replacing the CIEXYZ coordinates with the original data.

Segment maxima

The segment maxima GBD is provided in the sample implementation of GMAs provided by the CIE technical committee on gamut mapping (TC 8–03), making it a highly relevant basis for comparison. Segment maxima divides the color space into a number of segments around a center point. Each segment represents a uniform interval of spherical coordinates (polar and azimuth). For each segment, the color with the largest radius from the color space or gamut center is stored. These points can then be triangulated by taking advantage of the structure created by the use of uniform intervals.

Segment maxima, while theoretically a simple and straightforward algorithm, is not easily implemented in a way that provides optimal results. The basic algorithm is fast since it takes time linearly proportional to the number of input points, and requires little storage. All that is required is the coordinates of the colors with the largest radius per segment. However, the possibility of empty segments (segments that do not contain any measurement colors) generates the need for an interpolation algorithm. The source code provided by Morovič, available from the CIE Division 8 home page, provides a reference implementation of segment maxima as well as an intricate interpolation function. In this study, we have followed this implementation closely. Good interpolation is necessary to avoid artificial concavities caused by a mismatch between the uniform segment division and the data measurements. The creation of the surface triangles from the extreme points also result in added complexity since there are a number of special cases that need to be handled when triangulating the surface:

(i) The extreme points of four neighboring segments form a surface element consisting of a four-sided, nonplanar polygon. This can be divided into triangles in two different ways, depending on the choice of diagonal. However, due to the positioning of the points, sometimes one of the triangulations results in a triangle that faces inwards when viewed from the outside toward the gamut center, as illustrated by Figure 3(a). Implementations of the segment maxima GBD that create a surface structure consisting of triangles should avoid this folding of the gamut surface by selecting the other diagonal when such problems are detected. This results in the triangles in Fig. 3(b), where the dotted diagonal line correctly identifies a shared internal edge.

(ii) The bottom and the top of the gamut need special considerations when constructing a surface from the extreme points. This can be solved by adding an artificial top or bottom point, calculated from the surrounding data points. Alternatively, it is possible to perform a two-dimensional (2D) triangulation of the neighboring points based on their position in the plane perpendicular to the L-axis (or equivalent).

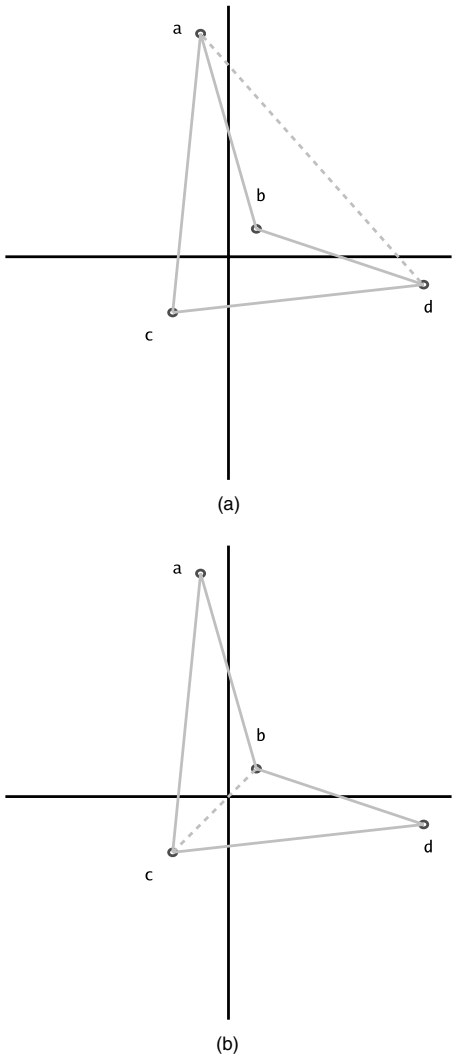

Figure 3. The two different triangulations of segment maxima points. The incorrect triangulation results in overlapping triangles. (a) Incorrect triangulation. (b) Correct triangulation.

Alpha Shapes

Edelsbrunner and Mücke[25] presented a technique for calculating approximations to shapes of a set of points in 3D. These approximations, named alpha shapes, are constructed from a 3D Delaunay triangulation of the point set, using a parameter α to determine which tetrahedra, triangles, edges, and points are part of the shape. The simplices having points such that there exists a sphere with a radius less than or equal to α containing these points are part of the specified alpha shape. When applied to measurement data, α should be large enough to ensure that the shape consists of one single part made up of interconnected simplices. Computationally, the complexity of the algorithm is defined by the Delaunay triangulation in n dimensions, which is equivalent to finding the convex hull of the same number of points in $n+1$ dimensions.

Uniform Segment Visualization

The uniform segment visualization[29] method is an attempt to combine the modified convex hull algorithm with a segment-based preprocessing step to reduce the number of points used to define the gamut surface. First, the color space is divided into segments using a sphere tesselation technique to create segments having a more uniform size than existing algorithms. Similar to segment maxima, for each segment the point with the largest radius is kept for further processing. The modified convex hull algorithm is used to construct a surface from these points.

EXPERIMENTAL METHOD

In order to perform objective evaluations of the performance of GBDs using our method, it is necessary to be able to construct a reference surface that contains all the data points and follows the data set closely. This surface can then

Table I. The device profiles used for generating test data.

Device	Device type
PAL/SECAM RGB	Generic
ColorMatch RGB	Generic
Adobe RGB (1998)	Generic
Apple RGB	Generic
Best RGB	Generic
Bruce RGB	Generic
Wide Gamut RGB	Generic
NTSC RGB	Generic
Philips 202P4	Monitor
Apple Studio Display 21″	Monitor
Sony STYLEPRO CPD-E240/B	Monitor
Hitachi CM821FET	Monitor
NEC MultiSync LCD1970GX	Monitor
Sony CPD-E530	Monitor
Canon iPF8000	Printer
Canon iPF5000	Printer
Epson Stylus Pro 10600	Printer
Epson Stylus Pro 4880	Printer
Epson Stylus Photo R2400	Printer

be used when analyzing the performance of each algorithm. We base our method on the use of device data and gamuts. The advantage of this approach is that these data sets, unlike images, have an internal structure that can be utilized in the creation of the reference gamut surface. We restrict the choice of devices to device types whose color spaces have three components, thereby further simplifying the task of constructing such a reference surface. Simulated data from a number of different RGB and CMY based devices can then be used to ensure the general validity of the results.

We have performed an evaluation of the performance of several different GBDs, using several different parameters and a selection of different data sets.

Devices and profiles

Nineteen ICC profiles were used as the basis for the experiment, providing the means to create the simulated measurement data. In order to test the performance of the algorithms on data from different devices, three different types of profiles have been included, as can be seen in Table I. The generic profiles include RGB working spaces like Adobe RGB. The monitor profiles include an assortment of monitor profiles, while the printer profiles include profiles for some Canon and Epson printers. All device profiles are generic profiles provided by the manufacturers.

All profiles have a three-component device color space, while only the printer profiles contain 3D lookup tables. We utilize a color management module (CMM) with the relative colorimetric option to generate simulated data points by converting colors from the device color space to CIELAB.

Data Sets

In order to evaluate the methods for construction of gamut boundary descriptors, each method is then applied to a variety of simulated data sets extracted from the given device gamut, and the resulting gamuts are compared against the reference. We use these general types of data sets:

(i) Data sets consisting of surface points. These data sets can be constructed using uniform sampling of each of the six sides of the color cube, eliminating shared points along the edges that have already been added to the set.

(ii) Data sets that in addition to surface points also include interior gamut points, typically found by utilizing uniform sampling along each of the three axes of the device color space.

(iii) Data sets based on standard test charts, e.g., TC 2.83 and TC 9.18 RGB test charts.

The effect of different measurement data was simulated by using the ICC profiles to transform data from the device color space to CIEXYZ values. The generated data points should all be on the inside of the device gamut and the data sets represent possible alternative bases for generating the gamut surfaces.

Metric/Baseline Truth GBD

The first step in our proposed method for GBD evaluation consists of constructing a reference gamut boundary for each of the devices that are to be part of the experiment. This surface is constructed by performing a dense sampling of the six sides of the three-dimensional RGB/CMY cube in the device color space, followed by transforming these data into the CIELAB color space.[12] The resulting points are then triangulated, creating a surface that closely follows the perfect gamut boundary of the device. It is necessary to inspect this surface to make sure that the devices do not exhibit behavior that causes this method to fail.[13,14] We have visually confirmed that the reference gamut of each device encloses all possible colors of that profile.

The surfaces can be evaluated by comparing a number of attributes against the reference surface. One such attribute is the gamut volume, which can be used as an indicator of, e.g., gamut overestimation. However, while the existence of a volume difference is indicative of a difference in the gamut boundaries, the opposite is not necessarily the case. There are obviously an infinite number of gamuts that may have the same volume, but whose surfaces are not equal.

This suggests the use of an alternative metric. The error introduced by a gamut boundary descriptor refers to the difference between the space contained by the GBD surface and the reference gamut. We introduce the concept of relative gamut mismatch, which refers to this difference volume (parts of the color space that are contained within exactly one of the gamuts and not the other) divided by the volume of the reference gamut,

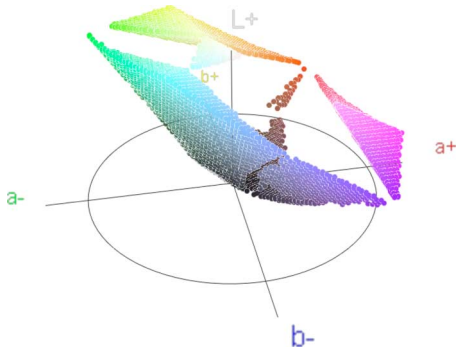

Figure 4. The voxels that represent the mismatch between 2 gamuts.

$$r_i = \frac{V(G_i \setminus G_{ref}) + V(G_{ref} \setminus G_i)}{V(G_{ref})}. \quad (2)$$

When comparing gamuts, it can be necessary to determine the union, intersection, and the difference between the two objects. In order to perform this operation fast, as well as independently of the structure of different GBDs, we employ a voxel based technique. Every gamut is represented by a 3D grid of binary values, where each value indicates whether the associated volume in the color space lies within the gamut boundary. The grid dimensions are chosen to take into account the perceptual scale along each axis, using the same subdivision as demonstrated by Giesen et al.[26] in CIELAB. The advantage of this data type is that each voxel only requires one bit of memory storage, and the difference between gamuts can be found easily by using an xor operation on the bits that represent two gamuts. Figure 4 displays the voxels that are the result of such an operation. This eliminates the need to compute the intersection of tetrahedra to compare the gamuts, and allows comparisons between gamuts represented by different structures. The construction of the grid depends only on a simple inside/outside test for each GBD type, and can thus be optimized by traversal of the gamut structure.

Choice of Algorithm Parameters

The classical convex hull and the convex hull in CIEXYZ are able to construct a surface without setting any additional parameters. When using the modified convex hull algorithm, some parameters for the preprocessing step can change the resulting gamut. First, one needs to decide on a center point that is used to calculate the radius of the points. We use the center of the CIELAB color space (50, 0, 0) as this point. The γ parameter defines the amount of nonlinear correction of the radius, in effect deciding the amount of concavities in the gamut surface. We include results from gamuts calculated using 0.05, 0.2, and 0.5 as the value of γ.

The alpha shapes algorithm defines a series of shapes, while the choice of alpha determines the final gamut. If $\alpha=\infty$, the alpha shape will be equal to the convex hull, while $\alpha=0$ results in just the disconnected input points. When applied to device measurement data, we only need to decide the minimum and maximum values of alpha for which each tetrahedron of the Delaunay triangulation should be included in the gamut object. Disconnected triangles, edges, and points are unsuitable for most gamut applications (e.g., gamut mapping).

Some optimization algorithms have been proposed for selecting a suitable alpha value, e.g., by finding the alpha which results in the largest connected group of tetrahedra without holes.[24] However, testing the connectivity of a graph built from the edge information of the tetrahedron structure is potentially slow. The graph state may switch back and forth between full connectivity and disconnected parts as the alpha value decreases, making it difficult to optimize the search for the correct alpha. Testing also shows that using this optimized alpha value is problematic, as the resulting gamut and the reference gamut can be quite different for some data sets. Based on this test, as well as empirical data showing that the convex hull is relatively consistent in its overestimation of the gamut volume, we choose the greatest alpha value satisfying the condition that the resulting alpha shape has a volume equal to or less than 90% of the convex hull. We refer to this algorithm as AS10% in the following figures, meaning alpha shape where 10% of the volume has been removed from the starting point of $\alpha=\infty$. Due to algorithm complexity and running time, we have not been able to calculate the alpha shapes for all the data sets.

The segment maxima algorithm depends on the selection of a center point. We use the color space center in our calculations, similar to our choice for the modified convex hull. We use 8, 12, and 16 subdivisions of the angles around this center point.

Data Analysis

Many methods that are currently in use for analyzing results depend on assumptions about the distributions of the underlying data. These assumptions, e.g., that results can be approximated using the normal distribution, and that algorithms have the same variance, cannot be made for GBD algorithms. In order to avoid this we have utilized statistical methods that do not make any assumptions about the distributions. We have used box plots to display the experimental data. By using this technique, we utilize the data directly and do not depend on the data following a specific statistical model. The plots show the median value as well as the upper and lower quartiles of the grouped data. The full range of the data is also plotted, along with any identified outliers.

We then use the sign test to test if there is a statistically significant difference between the performance of the algorithms. The nonparametric property of this test allows us to compute statistics without making any assumptions regarding the distribution of the results of the algorithms. The pairing of data points compensates for the difference in performance between the data sets and results in detection of significant differences even when plots of combined results do not appear to show any significant difference between algorithms.

While the performance of the algorithms can vary greatly, the probabilities derived from the sign test do not show the magnitude of the differences. Several algorithms have problems when they are applied to certain types of

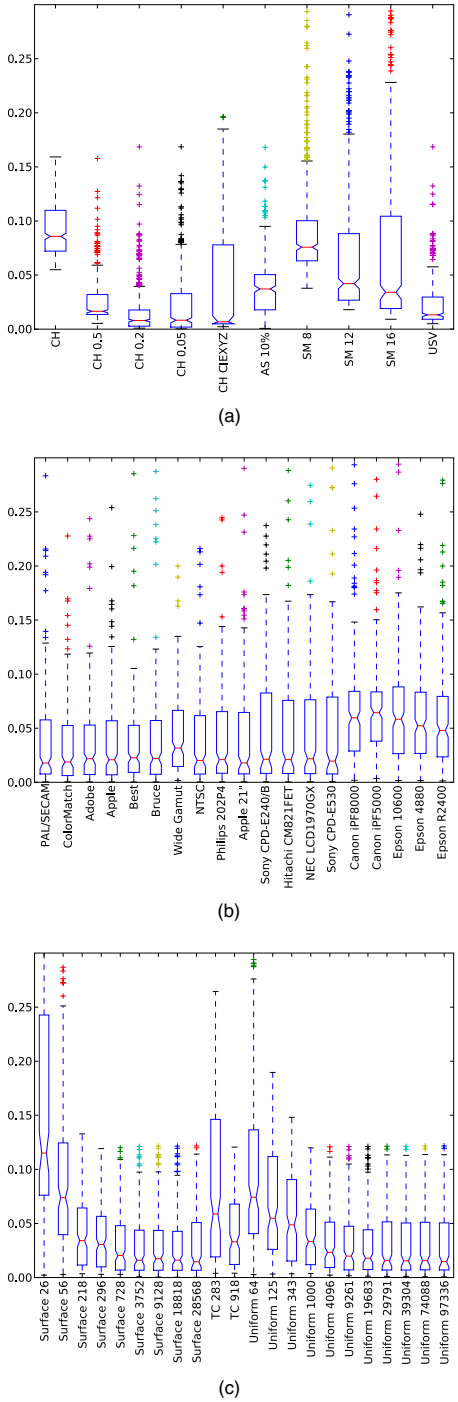

Figure 5. Overall results grouped by algorithm, device and data set. We see that the choice of data set and algorithm influences the accuracy of the constructed gamut boundaries. The algorithms perform slightly worse when used on data from printers than the other device types. (a) The performance of the GBD algorithms for all data sets. (b) The performance of all algorithms for each device. (c) The performance of all algorithms for the different data sets.

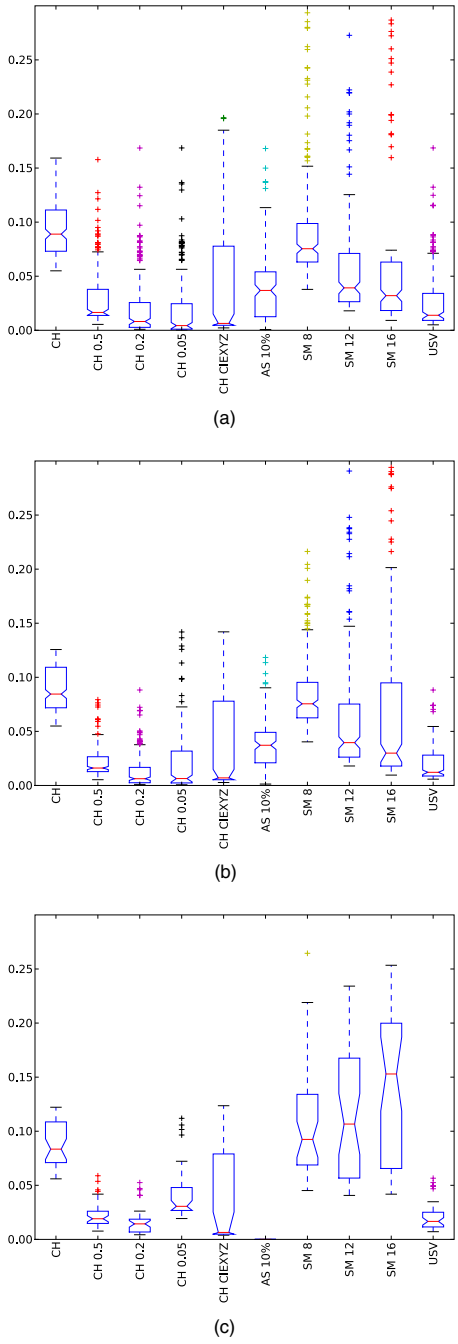

Figure 6. Results of the algorithms for the different types of data sets. (a) The performance of the algorithms for data sets containing surface points. (b) The performance of the algorithms for data sets containing internal points. (c) The performance of the algorithms for data sets from actual color charts.

data, while the sign test does not put any more weight to these results than to minor differences. Although the sign test can detect statistically significant differences in many situations, the box plot might reveal that the actual practical difference between the two algorithms is small.

RESULTS AND DISCUSSION

Overall Results

We have computed a total of 4047 gamut boundaries that have been compared with reference gamuts. The overall results for the different algorithms are presented in Figure 5(a). We see that the average performance and the variance of the algorithms vary greatly. While some algorithms generally perform well for almost all types of input, others do not give a satisfactory result for certain data sets. The convex

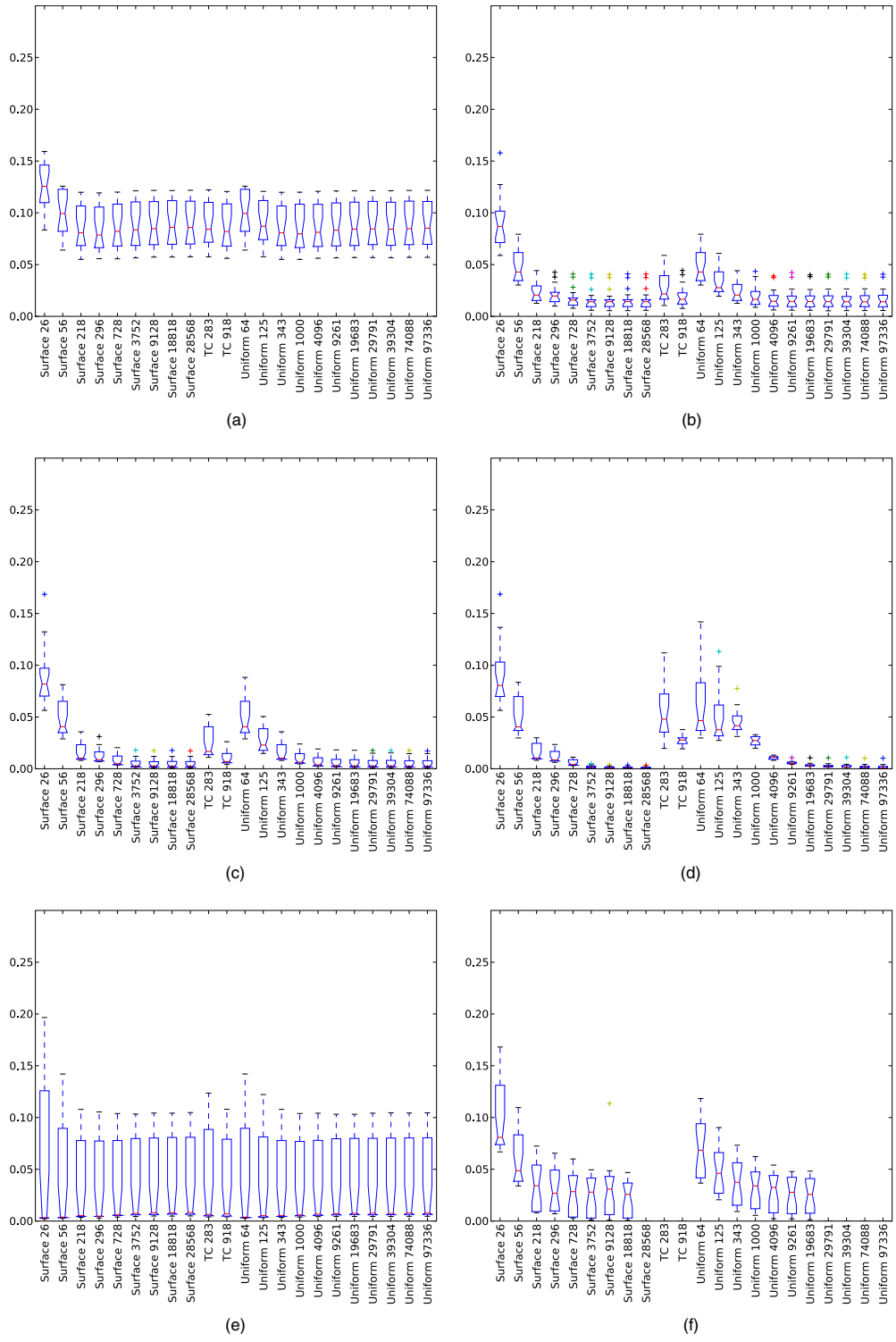

Figure 7. The performance of convex hull based algorithms for the different types of data sets (a) Convex hull. (b) Modified convex hull with gamma=0.5. (c) Modified convex hull with gamma=0.2. (d) Modified convex hull with gamma=0.05. (e) Convex hull computed in the CIEXZ color space. (f) Alpha shapes.

hull algorithm consistently overestimates the gamut size by close to 10%. The alpha shapes method performs better, but it is not as good as the most accurate algorithms. The modified convex hull algorithm generally results in accurate gamut boundaries. We see that using $\gamma=0.2$ gives both a better median gamut mismatch as well as a smaller variance than $\gamma=0.5$. Using an even smaller γ increases the variation without improving the median mismatch, making it less suitable for arbitrary data sets. The convex hull in CIEXYZ has a very small median mismatch, but the range of the results is relatively high. The segment maxima algorithm does not perform well. Although the average accuracy is improved when a higher number of segments is used, the variance increases dramatically.

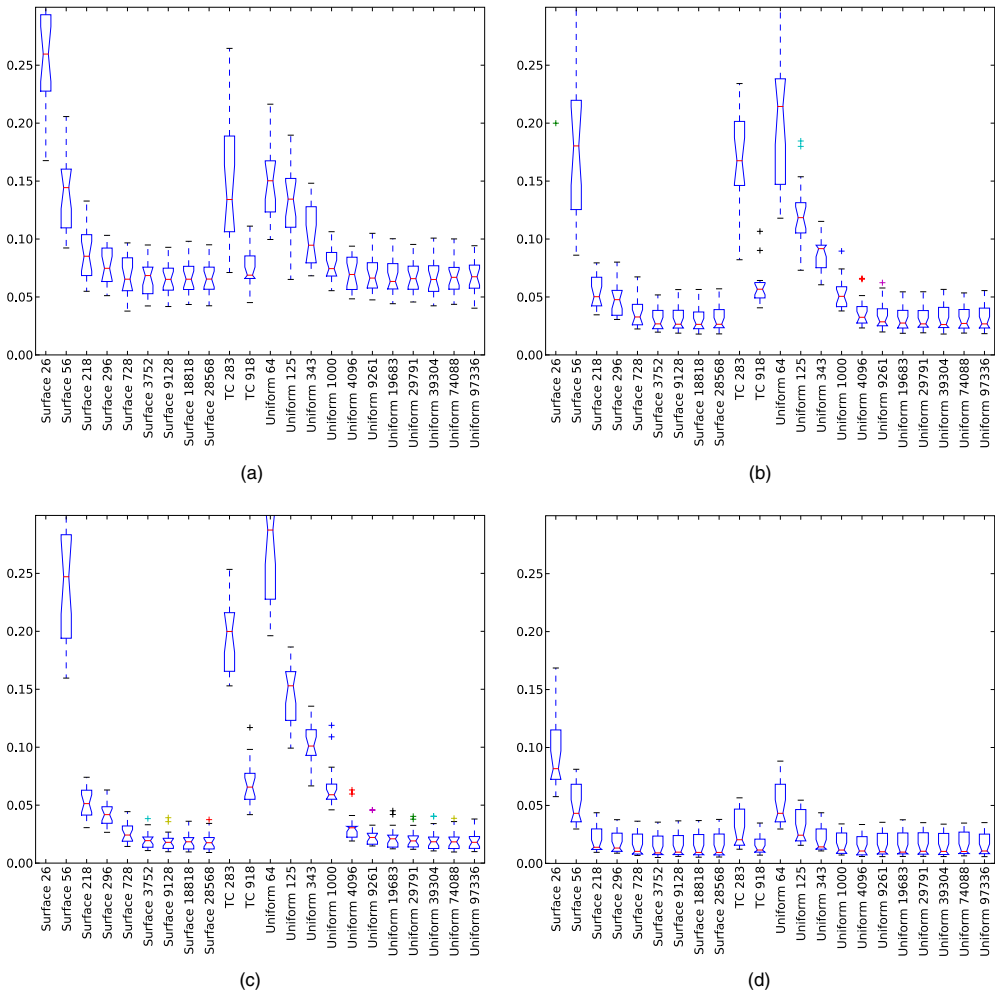

Figure 8. The performance of segment based algorithms for the different types of data sets. (a) Segment maxima with 8×8 subdivisions. (b) Segment maxima with 12×12 subdivisions. (c) Segment maxima with 16×16 subdivisions. (d) Uniform segment visualization.

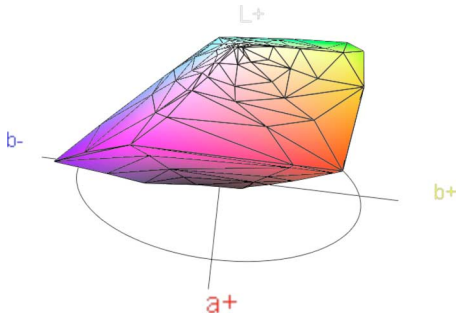

Figure 9. The segment maxima GBD.

Figure 5(b) shows that the performance of the algorithms is mostly independent of the device. However, the accuracy of the computed gamuts seems to be slightly worse for the printer profiles compared to the other profiles. Although the gamuts computed for data from printer profiles on average have a higher error than for the two other groups, the large in-class variance seen by grouping the results by the device type make it difficult to draw any clear conclusions. This tendency is in part caused by higher variation in sample density in CIELAB for printer profiles when the device color space is sampled uniformly. Also, the printer gamuts have a more irregular shape than the other device types based on simpler device models.

When we group the data by the different data sets, we see that the accuracy of the gamuts increases with a higher number of simulated measurement points, up to a certain limit where adding additional input points does not improve the performance of the algorithms. Clearly, a dense sampling of the gamut surface is preferable. Figure 5(c) also shows that using data sets containing only surface points reduces the number of points necessary to achieve a given level of gamut accuracy, e.g., 3752 surface points gives about the same results as using 19 683 points computed from a uniform subdivision of the device color space.

Algorithm Performance for the Different Data Sets

The selection of measurements used to calculate color gamuts is important, as we can see from the significant differences in gamut mismatch between algorithms applied to different types of measurement data in Figure 6. The inclusion of measurements within the gamut may reduce the accuracy of the resulting gamut since the GBDs, with the exception of conventional convex hull, may include some of the internal

Table II. A comparison of the gamut mismatch of the algorithms when used on data from the standard TC 9.18 test target. If the table cell contains $>$, this indicates that the gamut mismatch of the algorithm in the row is worse than the algorithm in the column, while $<$ indicates that the opposite relationship is true. A single $>$ or $<$ is used when the corresponding p-value is less than 0.05, while $\gg$ and $\ll$ indicate an even more significant p-value of less than 0.01

	CH	CH 0.5	CH 0.2	CH 0.05	CH CIEXYZ	SM 8	SM 12	SM 16	USV
CH		$\gg$	$\gg$	$\gg$	$>$	$\gg$	$\gg$		$\gg$
CH 0.5	$\ll$		$\gg$	$\ll$		$\ll$	$\ll$	$\ll$	$\gg$
CH 0.2	$\ll$	$\ll$		$\ll$		$\ll$	$\ll$	$\ll$	$\ll$
CH 0.05	$\ll$	$\gg$	$\gg$			$\ll$	$\ll$	$\ll$	$\gg$
CH CIEXYZ	$<$					$\ll$	$\ll$	$\ll$	
SM 8	$\ll$	$\gg$	$\gg$	$\gg$	$\gg$		$\gg$		$\gg$
SM 12	$\ll$	$\gg$	$\gg$	$\gg$	$\gg$	$\ll$		$\ll$	$\gg$
SM 16		$\gg$	$\gg$	$\gg$	$\gg$		$\gg$		$\gg$
USV	$\ll$	$\ll$	$\gg$	$\ll$		$\ll$	$\ll$	$\ll$	

points in the final gamut surface. It is therefore possible to optimize the performance of the GBD by selecting measurements specifically for the task of gamut boundary determination.

In the case of previously measured test charts [Fig. 6(c)], we see that the segment maxima algorithm does not perform well with such a small number of data points when internal points are included. It can be advantageous to apply a device characterization model, and use this to create artificial surface points. However, this approach will fail in cases where internal points in the device color space are part of the surface in CIELAB or a similar color space. Even if the transform from device color space to CIELAB does not distort the relationship between the positions of the data, it is still not possible to perform a straightforward trianglulation of the surface points for all devices. In particular, it is not possible to specify only surface points covering the entire CIELAB gamut of devices having a color space with more than three components, e.g., CMYK printers.

Individual Algorithm Performance

Figures 7 and 8 show the performance of the individual algorithms grouped by data set. The conventional convex hull technique overestimates the volume by a significant amount, but can be useful to identify colors that are guaranteed to be on the outside of the gamut boundary (given that the convex hull is based on a representative sampling of the device color space).

Results from the experiment, along with empirical evidence, suggest that the modified convex hull algorithm performs well on a wide variety of measurement data, assuming that the γ parameter is set to a sensible value. The results show that a choice of 0.2 as the value of γ gives accurate gamut boundaries for the data sets that have been tested. The modified convex hull algorithm shows a very stable performance, generally requiring few data points to generate an accurate gamut.

Computing the convex hull in the CIEXYZ color space results in a very small median gamut mismatch, but the results vary greatly. The somewhat unusual distribution of the results of the CIEXYZ convex hull algorithm can be explained by the convex nature of the gamuts of our monitor and generic device profiles in CIEXYZ. The large variance indicates that this algorithm does not work well for the printer gamuts because of their concavities in this color space. In fact, the algorithm performs worse than the normal convex hull in the case of the tested printer gamuts.

The alpha shapes algorithm does not perform as well as the modified convex hull. This algorithm has not been tested on the data sets containing a high number of points due to the computational complexity of the Delaunay triangulation. The alpha shapes algorithm also has the disadvantage that it is not as easy to implement as some of the better performing algorithms.

The segment maxima algorithm performs well on measurement data consisting of densely sampled surface points, where the use of a higher number of segments increases the quality of the gamut boundary without the risk of adding internal points. Additionally, this method results in a limited number of extreme points that can easily be specified by changing the number of segments, which is particularly suited to inclusion in file formats where size is important. One of the issues with the segment maxima GBD is that the surface points are positioned much closer together near the top and the bottom of the gamut, as seen in Figure 9. This is caused by the uniform subdivision of the spherical coordinate space into segments, and results in a need for additional sample points near the device white and black point to avoid unwanted artifacts in the gamut boundary.

The USV algorithm combines many of the advantages of a segment based method with the stable results of the modified convex hull. If a simpler surface consisting of fewer surface points is desired, this is clearly a better choice than the segment maxima method in terms of accuracy.

CONCLUSIONS

We have introduced a method for GBD evaluation, and looked at how several different GBDs perform on some data sets. The choice of GBD and parameter values influences the

surface that is constructed, and we have shown that it is important to make an informed choice based on the type of data and the requirements that apply to the situation.

The modified convex hull algorithm performs well on a range of different data using $\gamma=0.2$. This is clearly the method that should be used if there is no specific prior knowledge about the data that indicates that another algorithm would perform better. Table II shows that no other method performs better than this when looking at paired comparisons of the algorithms for the standard TC 9.18 measurement data. The convex hull in CIEXYZ is a very good choice if the device is known to have a convex gamut in this color space. The alpha shapes method does not perform as well as the modified convex hull for most data sets using our suggested α parameter, but has the possible advantage that with the right tool the α parameter can be adjusted interactively to give a pleasing visualization of the gamut for a specific data set. The uniform segment visualization algorithm does not follow densely sampled surface data as well as the modified convex hull due to its use of a reduced number of surface points. However, the surfaces constructed using USV show a consistent performance while avoiding the use of an excessive number of surface triangles and vertices.

ACKNOWLEDGMENTS

We would like to thank Professor Are Strandlie for constructive suggestions and discussions.

REFERENCES

[1] Ján Morovič and M. Ronnier Luo, "The fundamentals of gamut mapping: A survey", J. Imaging Sci. Technol. **45**, 283–290 (2001).
[2] David L. MacAdam, "Maximum visual efficiency of colored materials", J. Opt. Soc. Am. **25**, 361–367 (1935).
[3] Peter G. Engeldrum, "Computing color gamuts of ink-jet printing systems", Proc. SID, **27**, 25–30, 1986.
[4] Gary W. Meyer, Linda S. Peting, and Ferenc Rakoczi, "A color gamut visualization tool", in *Proceedings of IS&T and SID's Color Imaging Conference* (IS&T, Springfield, VA, 1993), pp. 197–201.
[5] Gary W. Meyer and Chad A. Robertson, "A data flow approach to color gamut visualization", in *Proceedings of IS&T and SID's 5th Color Imaging Conference* (IS&T, Springfiled, VA, 1997), pp. 209–214.
[6] H. E. J. Neugebauer, "Die theoretischen Grundlagen des Mehrfarbenbuchdrucks", Zeitschrift für wissenschaftliche Photographie, Photophysik und Photochemie **36**, 73–89 (1937).
[7] M. Mahy, "Calculation of color gamuts based on the Neugebauer model", Color Res. Appl. **22**, 365–374 (1997).
[8] Masao Inui, "Fast algorithm for computing color gamuts", Color Res. Appl. **18**, 341–348 (1993).
[9] Patrick G. Herzog, "Analytical color gamut representations", J. Imaging Sci. Technol. **40**, 516–521 (1996).
[10] Patrick G. Herzog, in *Specifying and Visualizing Colour Gamut Boundaries*, edited by Lindsay W. MacDonald and M. Ronnier Luo (Wiley, New York, 1998), Chap. 13, pp. 233–252.
[11] M. C. Stone, William B. Cowan, and J. C. Beatty, "Color gamut mapping and the printing of digital color images", ACM Trans. Graphics **7**, 249–292 (1988).
[12] P. G. Herzog, "Specifying and visualizing colour gamut boundaries", in *Color Imaging: Vision and Technology*, edited by L. W. MacDonald and M. R. Luo(Wiley, New York, 1998), Chap. 13, pp. 233–252.
[13] Jon Y. Hardeberg and Francis Schmitt, "Color printer characterization using a computational geometry approach", in *Proceedings of IS&T and SID's 5th Color Imaging Conference* (IS&T, Springfield, VA, 1997), pp. 96–99; Also in *Recent Progress in Color Management and Communications*, edited by R. Buckley (IS&T, Springfield, VA, 1998), pp. 88–91.
[14] Jon Y. Hardeberg, *Acquisition and Reproduction of Color Images: Colorimetric and Multispectral Approaches*. Dissertation.com, Parkland, Florida, 2001.
[15] C. Bradford Barber, D.P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls", ACM Trans. Math. Softw. **22**, 469–483 (1996).
[16] Xianfeng Zhao, Implementing an ICC printer profile visualization software. MSc thesis, School of Printing Management and Sciences in the College of Imaging Arts and Sciences of the Rochester Institute of Technology, February 2001.
[17] J. A. Stephen Viggiano and William J. Hoagland, "Colorant selection for six-color lithographic printing", in *Proceedings of IS&T and SID's 6th Color Imaging Conference* (IS&T, Springfield, VA, 1998), pp. 112–115.
[18] Raja Balasubramanian and Edul Dalal, "A method for quantifying the color gamut of an output device", in *Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts II*, Proc. SPIE Vol. **3018**, 110–116 1997.
[19] Gustav J. Braun and Mark D. Fairchild, "Techniques for gamut surface definition and visualization", in *Proceedings of IS&T and SID's 5th Color Imaging Conference* (IS&T, Springfield, VA, 1997), pp. 147–152.
[20] Richard L. Reel and Michael A. Penrod, "Gamut visualization tools and metrics", in *Proceedings of IS&T and SID's 7th Color Imaging Conference* (IS&T, Springfield, VA, 1999), pp. 247–251.
[21] Ján Morovič and M. Ronnier Luo, "Gamut mapping algorithms based on psychophysical experiment", in *Proceedings of IS&T and SID's 5th Color Imaging Conference* (IS&T, Springfield, VA, 1997), pp. 44–49.
[22] Ryoichi Saito and Hiroaki Kotera, "Extraction of image gamut surface and calculation of its volume", in *Proceedings of IS&T and SID's 8th Color Imaging Conference* (IS&T, Springfield, VA, 2000), pp. 330–334.
[23] Ryoichi Saito and Hiroaki Kotera, "Image-dependent three-dimensional gamut mapping using gamut boundary descriptor", J. Electron. Imaging **13**, 630–638 (2004).
[24] Tomasz J. Cholewo and Shaun Love, "Gamut boundary determination using alpha-shapes", in *Proceedings of IS&T and SID's 7th Color Imaging Conference* (IS&T, Springfield, VA, 1998), pp. 200–204.
[25] Herbert Edelsbrunner and Ernst P. Mücke, "Three-dimensional alpha shapes", ACM Trans. Graphics **13**, 43–72 (1994).
[26] Joachim Giesen, Eva Schubert, Klaus Simon, and Peter Zolliker, "Toward image-dependent gamut mapping, fast and acccurate gamut boundary determination", Proc. SPIE, Vol. **5667**, 201–210, 2005.
[27] Ivar Farup, Jon Y. Hardeberg, Arne M. Bakke, Ståle Kopperud, and Anders Rindal, "Visualization and interactive manipulation of color gamuts", in *Proceedings of IS&T and SID's 10th Color Imaging Conference* (IS&T, Springfield, VA, 2002), pp. 250–255.
[28] Karl Guyler, "Visualization of expanded printing gamuts using 3-dimensional convex hulls", American Ink Maker **79**, 36–56 (2001).
[29] A. M. Bakke, I. Farup, and J. Y. Hardeberg, "Improved gamut boundary determination for color gamut mapping", in *Advances in Printing Science and Technology* (International Association of Research Organizations for the Information, Media, and Graphics Arts Industries, Darmstadt, Germany, 2008), Vol. **35**, pp. 365–372.