

Evaluation of Gamut Boundary Descriptors

Arne M. Bakke, Jon Y. Hardeberg, and Ivar Farup
Gjøvik University College
Gjøvik, Norway

Abstract

Several techniques for the computation of gamut boundaries have been presented in the past. In this paper 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. Preliminary results from experiments using several gamut boundary descriptors are presented, and the performance of the different algorithms is discussed.

Introduction

The construction of a gamut boundary descriptor (GBD) is the first step in the process of performing gamut mapping. While there has been done extensive [1] research 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.

Gamut Boundary Descriptors

In order to perform gamut mapping, the boundary of the color gamut needs to be determined. There are several known approaches to this problem. 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 to 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 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 3 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]. One issue with this approach is that device gamut measurements generally do not constitute completely convex objects in CIELAB or related color spaces. Guyler [17] proposed that the convex hull algorithm be performed in CIEXYZ, and the resulting surface transformed to the desired color space. Guyler argues that color data tend to be more convex in the CIEXYZ space, and that this ap-

proach thus results in a better approximation. Similarly, if the data originates from a reasonably well-behaved printer, it is possible to compute the convex hull in a linearized dye density space [18].

Balasubramian and Dalal [19] presented a different improvement of the standard convex hull method. By introducing a non-linear method as a pre-processing step before the convex hull is found, the gamut surface is given the ability to follow concavities in the original data set. Before the convex hull is computed, the color data is pre-processed by a gamma function based on the distance from the color to the color space center, and a parameter, γ . By using a γ between 0 and 1, the data is 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.

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 [20, 21]. Segment maxima [22] is a related method that performs a subdivision of the color space into segments based on the polar coordinates of the colors. 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. The segment maxima technique has also been applied to image data [23, 24], where the mass center can be used as the origin of the spherical coordinate system.

Cholewo and Love [25] used alpha shapes [26] to find the gamut boundary of both devices and images. An alpha shape is constructed from a 3D Delaunay triangulation of the point set, using an α parameter to determine which tetrahedra, triangles, edges and points are a part of the shape. The α parameter should be large enough to ensure that the shape consists of one single part made up of interconnected simplices, and Cholewo and Love suggest that the optimal α should be found by interactively changing the shape.

Giesen et al. [27] 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 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.

Method

Evaluation of GBDs

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

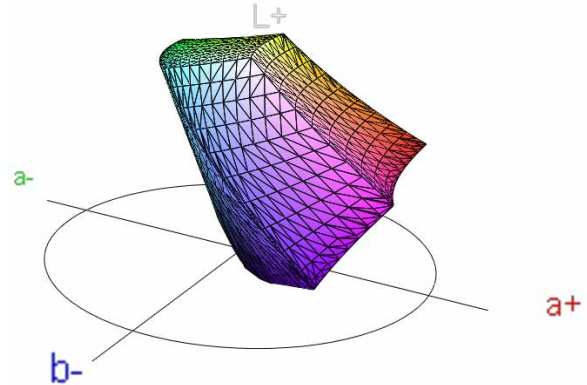


Figure 1. A gamut surface found by using the GBD proposed by Balasubramian and Dalal.

device types whose color space have 3 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.

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 6 sides of the 3-dimensional RGB/CMY cube in the device color space, followed by transforming these data into the CIELAB color space. The resulting points are then triangulated, creating a surface that closely follows the perfect gamut boundary of the device. It is necessary to visually inspect this surface to make sure that the devices do not exhibit behavior that causes this method to fail [13, 14].

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

- Data sets consisting of surface points. These data sets can be constructed using uniform sampling of each of the 6 sides of the color cube, eliminating shared points along the edges that have already been added to the set.
- Data sets that in addition to surface points also include interior gamut points, typically found by utilizing uniform sampling along each of the 3 axes of the device color space.
- Data sets based on standard test charts, e.g., TC 2.83 and TC 9.18 RGB test charts.

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

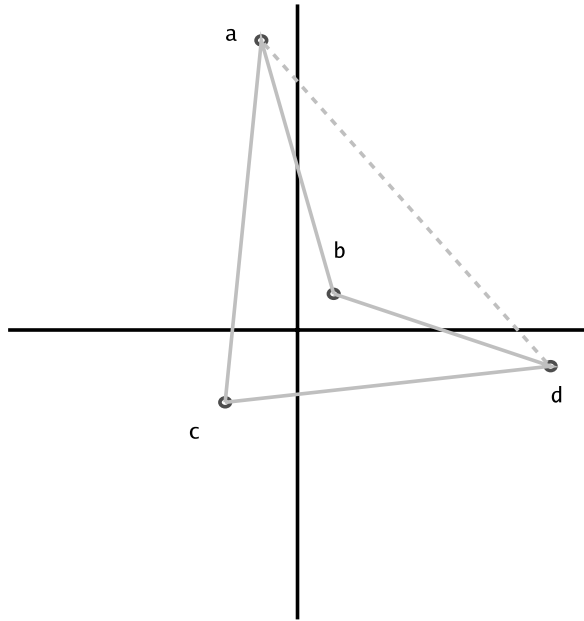


Figure 2. Incorrect triangulation of segment maxima extreme points.

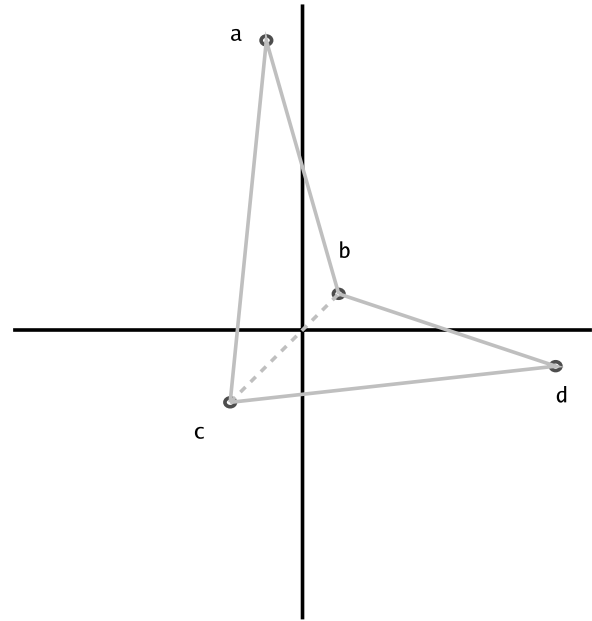


Figure 3. Correct triangulation of segment maxima extreme points.

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.

Framework and implementation

We have implemented and tested our method by using the ICC3D [28, 29] application as a basic framework for implementation of our algorithm. 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.

The modified convex hull algorithm is based on the improvements suggested by Balasubramian and Dalal [19], resulting in surfaces similar to the one illustrated in Figure 1.

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 Morovič¹, available from the CIE 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:

- The extreme points of 4 neighboring segments form a surface element consisting of a 4-sided, non-planar polygon. This can be divided into triangles in 2 different ways, depending on the

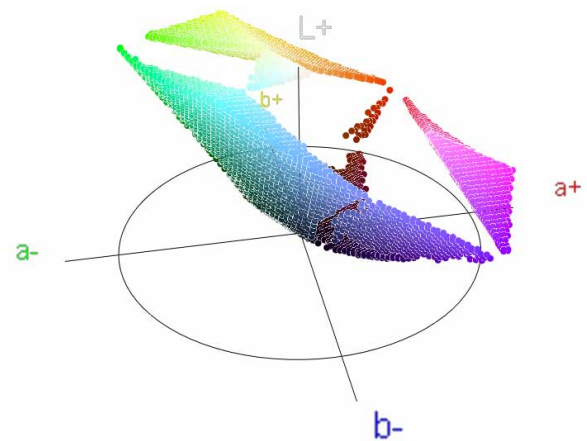


Figure 4. The voxels that represent the mismatch between 2 gamuts.

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 towards the gamut center, as illustrated by Figure 2. 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 Figure 3, where the dotted diagonal line correctly identifies a shared internal edge.

- 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 2D triangulation of the neighboring points based on their position in the plane perpendicular to the L-axis (or equivalent).

When comparing gamuts, it can be necessary to determine the

¹<http://www.colour.org/tc8-03/pgma>

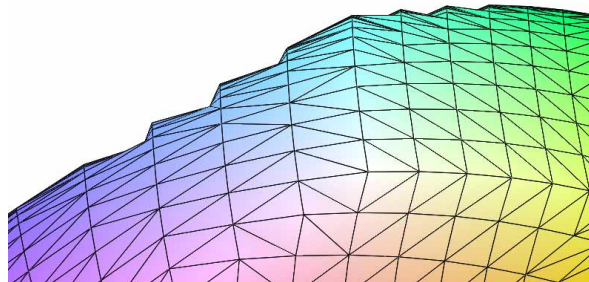


Figure 5. The use of a small γ value in the pre-processing step has caused artifacts in the surface.

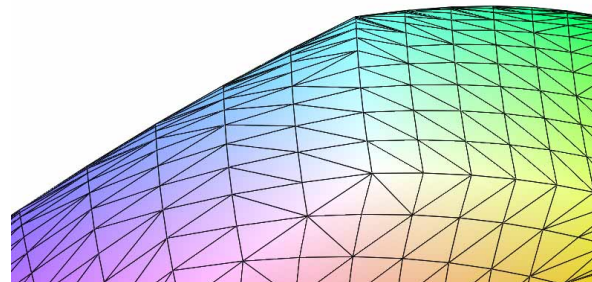


Figure 6. The same gamut as in Figure 5 but pre-processed using a larger value for γ .

union, intersection, and the difference between the 2 objects. In order to perform this operation fast, as well as independent 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, as demonstrated by Giesen et al. [27]. The advantage of this data type is that each voxel only requires 1 bit of memory storage, and the difference between gamuts can be found easily by using an xor operation on the bits that represent 2 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.

Preliminary results and discussion

We have performed an evaluation of the performance of 3 different GBDs, using several different parameters and a selection of different data sets. The results are displayed in Table 1, where the average relative gamut mismatch for each GBD is displayed per data set. 20 ICC profiles were used as the basis for the experiment, providing the means to create the simulated measurement data.

The effect of different measurement data was simulated by using the ICC profiles to transform data from the device color space to CIE XYZ values. The data sets are divided into 3 main categories, depending on their distribution in the device color space:

- Uniform sampling of the device color space. This creates a set of data points that include internal gamut points as well as surface data, denoted in the table by the number of subdivisions per axis.
- Uniform sampling of the surface of the device color space cube. The conversion from the device color space to CIELAB results in a number of theoretical gamut surface points, where the number of points is given in the table.
- Device data from RGB color charts, typically used to profile devices. We have used the TC 2.83 and TC 9.18 RGB test charts.

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 2 important considerations that make such an assumption false:

- 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.
- 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 5 and 6 illustrate this effect.

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. Clearly, a dense sampling of the gamut surface is preferable. 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 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, 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.

Results from the experiment, along with empirical evidence, suggests 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 table of results shows that a choice

Table 1. Results: Average relative gamut mismatch for a selection of segment maxima and convex hull based GBDs.

Data set	CH	CH $\gamma = 0.5$	CH $\gamma = 0.2$	CH $\gamma = 0.05$	SM 8	SM 12	SM 16
Uniform 4x4x4	0.1226797	0.0403273	0.0406264	0.0465822	0.1673969	0.3210610	0.2882731
Uniform 5x5x5	0.1106839	0.0264459	0.0226717	0.0368184	0.1415463	0.1305931	0.1529657
Uniform 7x7x7	0.1066282	0.0204723	0.0102405	0.0504724	0.1325726	0.0942938	0.0961941
Uniform 10x10x10	0.1083067	0.0162491	0.0064134	0.0311609	0.0876852	0.0495478	0.0699493
Uniform 16x16x16	0.1081659	0.0143136	0.0037478	0.0115336	0.0819844	0.0371169	0.0316712
Uniform 21x21x21	0.1095383	0.0145336	0.0029736	0.0052169	0.0757557	0.0281962	0.0222578
Uniform 27x27x27	0.1102861	0.0142432	0.0024281	0.0025513	0.0745768	0.0274748	0.0210613
Uniform 31x31x31	0.1110046	0.0141901	0.0024808	0.0019618	0.0759736	0.0267263	0.0197852
Uniform 24x34x34	0.1101952	0.0140933	0.0024457	0.0015659	0.0755866	0.0273948	0.0209728
Uniform 42x42x42	0.1112822	0.0139971	0.0022346	0.0007742	0.0744902	0.0272993	0.0194517
Uniform 46x46x46	0.1109479	0.0139356	0.0022346	0.0006246	0.0750708	0.0293843	0.0194869
Surface 26	0.1459200	0.0881098	0.0875291	0.0811244	0.2427308	0.4894646	0.6265341
Surface 56	0.1226851	0.0403466	0.0406897	0.0397748	0.1610786	0.3058989	0.2602120
Surface 218	0.1066555	0.0205076	0.0103022	0.0103022	0.0987551	0.0512471	0.0621035
Surface 296	0.1055558	0.0199974	0.0088242	0.0085251	0.0925351	0.0565697	0.0490564
Surface 728	0.1083789	0.0158447	0.0049267	0.0042229	0.0793113	0.0332289	0.0296395
Surface 3752	0.1105872	0.0140147	0.0025689	0.0010469	0.0719388	0.0269034	0.0219063
Surface 9128	0.1108335	0.0141467	0.0023050	0.0007126	0.0747629	0.0266307	0.0195397
Surface 18818	0.1117309	0.0140675	0.0021906	0.0005279	0.0764257	0.0276600	0.0188007
Surface 28568	0.1112294	0.0140939	0.0021202	0.0003695	0.0750004	0.0275192	0.0188623
TC 9.18	0.1085608	0.0170935	0.0074163	0.0316181	0.0848604	0.0618814	0.0793004
TC 2.83	0.1099430	0.0218619	0.0173664	0.0503572	0.1768924	0.1809129	0.2089506

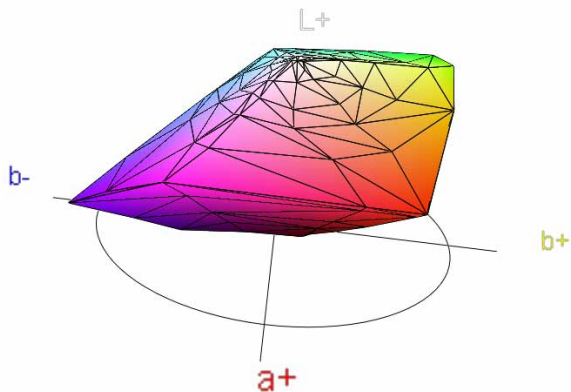


Figure 7. The segment maxima GBD.

of 0.2 as the value of γ results in a method that performs well for the data sets that have been tested. 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).

Segment maxima can perform 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, the segment maxima 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 7. 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.

Conclusion and Future Work

We have introduced a method for GBD evaluation, and looked at how the segment maxima, convex hull and modified convex hull GBDs perform on some data sets. The choice of GBD and its parameter values influences the surface that is constructed, and we have shown that the modified convex hull algorithm performs well on a range of different data given a correct choice of parameter.

The evaluation method presented here can easily be applied to different GBDs due to its voxel based comparison approach, which enables the inclusion of several other GBDs in future work. Alpha shapes using a method for automatic selection of α is one of the GBDs that are suitable for such work, as well as the recently introduced discrete flow complex.

References

- [1] Ján Morovič and M. Ronnier Luo. The fundamentals of gamut mapping: A survey. *Journal of Imaging Science and Technology*, 45(3):283–290, 2001.
- [2] D. L. MacAdam. Maximum visual efficiency of colored materials. *J. Opt. Soc. Amer.*, 25:361–367, 1935.
- [3] P. G. Engeldrum. Computing color gamuts of ink-jet printing systems. In *SID Proceedings*, volume 27, pages 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: Transforms and Transportability of Color*, pages 197–201, Scottsdale, Arizona, November 1993.
- [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: Color Science, Systems and Applications*, pages 209–214, Scottsdale, Arizona, 1997.
- [6] H. E. J. Neugebauer. Die theoretischen Grundlagen des Mehrfarbendruckes. *Zeitschrift für wissenschaftliche Photographie, Photo-physik und Photochemie*, 36(4):73–89, 1937.
- [7] M. Mahy. Calculation of color gamuts based on the Neugebauer model. *Color Research and Application*, 22:365–374, 1997.
- [8] Masao Inui. Fast algorithm for computing color gamuts. *Color Research and Applications*, 18(5):341–348, October 1993.
- [9] Patrick G. Herzog. Analytical color gamut representations. *Journal of Imaging Science and Technology*, 40:516–521, 1996.
- [10] Patrick G. Herzog. Specifying and visualizing colour gamut boundaries. In L. W. MacDonald and M. R. Luo, editors, *Colour Imaging: Vision and Technology*. John Wiley & Sons Ltd., 1998.
- [11] M. C. Stone, W. B. Cowan, and J. C. Beatty. Color gamut mapping and the printing of digital color images. *ACM Transactions on Graphics*, 7(4):249–292, 1988.
- [12] S. B. Bolte. A perspective on non-impact printing in color. In *Color Hardcopy and Graphic Arts*, volume 1670 of *Proc. SPIE*, pages 1–11, 1992.
- [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: Color Science, Systems and Applications*, pages 96–99, Scottsdale, Arizona, 1997. Also in R. Buckley, ed., *Recent Progress in Color Management and Communications*, IS&T, pages 88–91, 1998.
- [14] Jon Y. Hardeberg. *Acquisition and Reproduction of Color Images: Colorimetric and Multispectral Approaches*. Dissertation.com, Parkland, Florida, USA, 2001.
- [15] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.
- [16] Xianfeng Zhao. Implementing an ICC printer profile visualization software. Master's thesis, School of Printing Management and Sciences in the College of Imaging Arts and Sciences of the Rochester Institute of Technology, February 2001.
- [17] Karl Guyler. Visualization of expanded printing gamuts using 3-dimensional convex hulls. *American Ink Maker*, 79(9):36–56, September 2001.
- [18] 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: Color Science, Systems and Applications*, volume 6, pages 112–115, Scottsdale, Arizona, 1998.
- [19] 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*, volume 3018 of *Proc. SPIE*, San Jose, CA, January 1997.
- [20] 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: Color Science, Systems and Applications*, pages 147–152, Scottsdale, Arizona, 1997.
- [21] Richard L. Reel and Michael A. Penrod. Gamut visualization tools and metrics. In *Proceedings of IS&T and SID's 7th Color Imaging Conference: Color Science, Systems and Applications*, pages 247–251, Scottsdale, Arizona, November 1999.
- [22] Ján Morovič and M. R. Luo. Gamut mapping algorithms based on psychophysical experiment. In *Proceedings of IS&T and SID's 5th Color Imaging Conference: Color Science, Systems and Applications*, pages 44–49, Scottsdale, Arizona, 1997.
- [23] Ryoichi Saito and Hiroaki Kotera. Extraction of image gamut surface and calculations of its volume. In *Color Imaging Conference*, pages 330–334, 2000.
- [24] Ryoichi Saito and Hiroaki Kotera. Image-dependent three-dimensional gamut mapping using gamut boundary descriptor. *Journal of Electronic Imaging*, 13(3):630–638, 2004.
- [25] Tomasz J. Cholewo and Shaun Love. Gamut boundary determination using alpha-shapes. In *Proceedings of IS&T and SID's 7th Color Imaging Conference: Color Science, Systems and Applications*, pages 200–204, 1999.
- [26] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994.
- [27] Joachim Giesen, Eva Schubert, Klaus Simon, and Peter Zolliker. Toward image-dependent gamut mapping: fast and accurate gamut boundary determination. In *Color Imaging X: Processing Hardcopy, and Applications*, pages 201–210, San Jose, CA, USA, 2005.
- [28] Ivar Farup, Jon Y. Hardeberg, Arne M. Bakke, Stle Kopperud, and Anders Rindal. Visualization and interactive manipulation of color gamuts. In *Proceedings of IS&T and SID's 10th Color Imaging Conference: Color Science and Engineering: Systems, Technologies, Applications*, pages 250–255, Scottsdale, Arizona, 2002.
- [29] Ivar Farup and Jon Y. Hardeberg. Interactive color gamut mapping. In *The 11th International Printing and Graphics Arts Conference*, Bordeaux, France, October 2002.

Author Biography

Arne Magnus Bakke received his MSc in Media Technology at Gjøvik University College (Gjøvik, Norway) in collaboration with the Royal Institute of Technology (Stockholm, Sweden) in 2004. He has previously done research on the visualization and mapping of image and device gamuts both in colorimetric and spectral representations. He is the principal programmer of the ICC3D software (<http://www.colorlab.no/icc3d>).