Improved gamut boundary determination for colour gamut mapping

Arne M. Bakke, Ivar Farup, Jon Y. Hardeberg

The Norwegian Color Research Laboratory, Gjøvik University College
P.O. Box 191, N-2802 Gjøvik, Norway
E-mail: arne.magnus.bakke@hig.no

Abstract

We propose a new method for the computation of gamut boundaries, consisting of a combination of the segment maxima gamut boundary descriptor, the modified convex hull algorithm, and a sphere tessellation technique. This method gives a more uniform subdivision of the colour space into segments, and thus a more consistent level of detail over the gamut surface. First, the colour space is divided into segments around a centre point using the triangles from the tessellation algorithm. The measurement points are processed, and the point with the largest radius is found for each non-empty segment. The convex hull algorithm with a pre-processing step is then applied to these maxima points to generate the final gamut surface. The method is tested on different input data, including data sets both with and without internal gamut points. Different numbers of segments are used, and the resulting gamut boundaries are compared with the gamuts constructed using the segment maxima method. A reference gamut is constructed for each device, and the average mismatch is calculated. Our method is shown to perform better than the segment maxima method, particularly for a higher number of segments.

Keywords: gamut boundary, convex hull, segment maxima, sphere tessellation

1. Introduction and background

The colour gamut of a device is the range of colours that the device can reproduce. When an image is to be reproduced on another device, it is necessary to apply a gamut mapping algorithm (GMA) to compensate for the differences in their colour gamuts. These algorithms use gamut boundary descriptors (GBDs) to represent the extents of the colour gamuts. In order for the GMAs to give good results, it is important that the GBD is as accurate as possible. Any inaccuracies might result in the GMA failing to move all points to the inside of the destination gamut, or unnecessarily drastic gamut clipping.

Several methods exist that can be used to determine the boundary of a colour gamut with varying degree of success. Some algorithms take advantage of existing knowledge about the colorimetric behaviour of a device, and can only be used to compute gamut boundaries for a certain device type. By implementing a device model that is used to characterise the device, the theoretical boundaries of the gamuts are determined by the parameter limits used in the model, e.g., ink coverage.

GBDs based on device models

MacAdam introduced one of the earliest attempts to create an analytical approximation of a device gamut (MacAdam, 1935). Later, the gamuts of printing systems were determined indirectly using the Kubelka-Munk (Engeldrum, 1986) equations. Meyer et al. (Meyer et al., 1993, Meyer and Robertson, 1997) showed how to use this approach. Alternatively, Mahy found the gamut of a multi-ink printing system using the Neugebauer equations (Neugebauer, 1937). Inui made the assumption that there is a correspondence between colour space and the dye amount space (Inui, 1993), and used this to determine the gamut. Herzog used an analytical, mathematical approach to describe the colour gamut of a device, using deformed cubes referred to as gamulyts (Herzog, 1996, Herzog, 1998). A set of distortion functions is defined, and used to fit the deformed cube to the colour gamut. By extending this model, it can also be used for devices with more than 3 colorants.

Point based methods

When no specific characterization model of the device can be assumed, it is necessary to utilize alternatives to the analytical models. Usually, measured or simulated data points from the device are used as input to these algorithms. By processing a set of points in a colour space without taking any structure into account, some of these algorithms can also be used to find the gamuts of images.

The colour gamut of a device with three colorants can be determined by assuming that the gamut boundary of a device is preserved between the device dependent and device independent colour spaces. By measuring colours that are part of the gamut boundary in the device dependent colour space, an estimate of the colour gamut can be found. Depending on the need for detail, planes (Stone et al., 1988) can be used to represent the boundary by connecting the theoretical extreme points. If better accuracy is required, Bolte (Bolte, 1992) suggested an approach where colours that form the basis of a regular structure in the device colour space are printed and measured. Assuming that the relative position of the points remains the same in the device independent colour space, a direct triangulation can be performed on the measured colours.

A possible complication of this method is that characteristics of the printing process and the colour space transformation can lead to the order of the triangle vertices being reversed, and internal points in the device colour space structure may end up outside the triangles that form the surface of the regular structure. This method can be further improved by checking for mirrored tetrahedra in the device independent colour space, and testing that there are no points on the outside of the surface structure (Hardeberg and Schmitt, 1997, Hardeberg, 2001). Guyler (Guyler, 2007) has proposed an extension of the direct triangulation approach for data from CMYK printers, using Delaunay triangulation to find the surface triangles where the determination of the gamut in the device dependent colour space is made difficult by the use of the black colorant.

If the only available information about the device is a number of points in a device-independent colour space, which have been determined to be reproducible by the device, there are still algorithms that are capable of determining the gamut boundary. The convex hull is one of several algorithms that can be used to calculate a surface that approximately encloses the data points (Meyer et al., 1993, Meyer and Robertson, 1997). Typically, the quickhull (Barber et al., 1996) algorithm is applied to the set of input points. Zhao used quickhull to determine the gamut boundaries of ICC profiles (Zhao, 2001). One issue with this approach is that device gamut data generally do not constitute completely convex objects in CIELAB or related colour spaces.

Guyler (Guyler, 2001) converted the input points to the CIEXYZ colour space before using Qhull to compute the convex hull. The vertices of the surface could then be transformed to CIELAB, resulting in a gamut that more closely follows the concavities of the data. Guyler claims that the boundary of measurement data from a printer tend to be more convex in the CIEXYZ space, which results in a better approximation by using this approach. He also computes correction factors to compensate for the overestimation of gamut volume introduced by using the convex hull. An alternative to using CIEXYZ is to compute the convex hull in a linearised dye density space (Viggiano and Hoagland, 1998), assuming that the printing process does not distort the relationship between the points when performing the subsequent transformation to CIELAB.

Balasubramanian and Dalal extended the convex hull with a pre-processing step (Balasubramanian and Dalal, 1997) to better follow the concavities of the gamut. They suggested that the data points should be changed using a non-linear gamma function based on the distance from the colour to a centre point within the gamut, before the convex hull is calculated from the altered points. By varying the γ parameter from 1 to 0, the detail level of the gamut can be increased by making the object more concave. A γ value of 1 leaves the points unchanged before the convex hull is applied, while smaller parameter values make the pre-processed data more convex. With an optimal choice of γ, the final gamut boundary will closely follow the perceived surface of the data points (Bakke et al., 2006). Figure 1 shows a gamut boundary computed using this method.

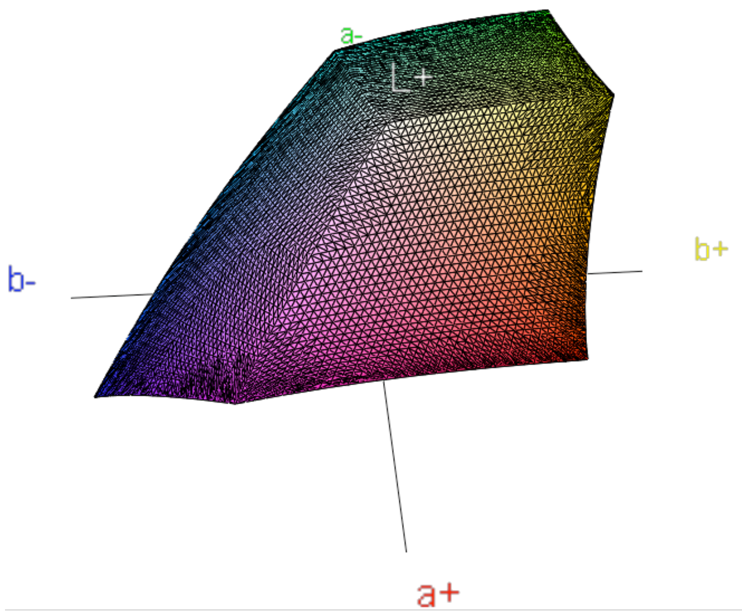

Figure 1. A gamut surface created using Balasubramanian and Dalal modified convex hull.

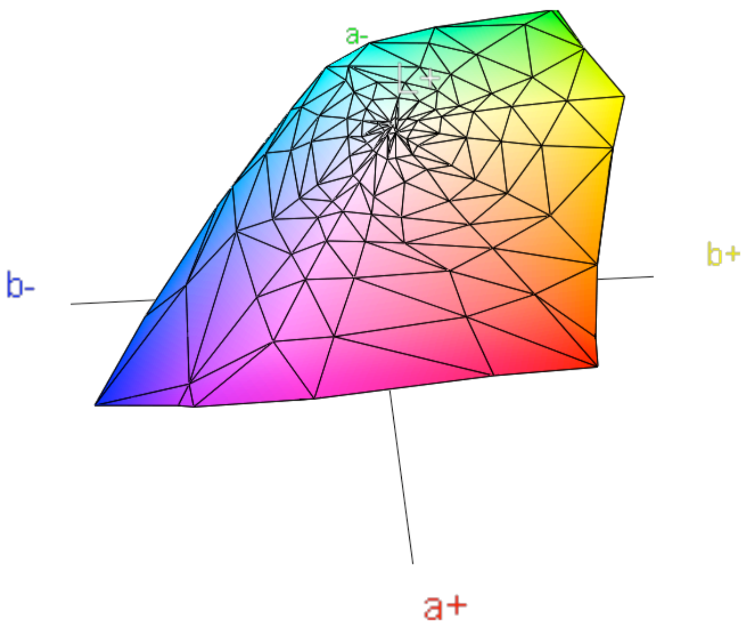

Figure 2. A gamut surface created using segment maxima. Note the smaller triangles near the white point.

By dividing the colour space into cells based on the lightness and hue, a different method to find the gamut boundary can be defined (Braun and Fairchild, 1997, Reel and Penrod, 1999). The colour with the largest chroma value is determined for each cell, and later used as vertices in a regular triangulation of this structure. Segment maxima (Morovic and Luo, 1997) is a similar method, which divides the colour space based on spherical coordinates. The colour that has the largest radius in its segment is found, and simulated points are calculated using interpolation for empty segments. This method can also be used to find the gamuts of colour images (Saito and Kotera, 2000, Saito and Kotera, 2004), substituting the mass centre for the colour space centre as the origin of the spherical coordinate system. The segment maxima method results in a gamut where the surface is sampled more densely near the top and the bottom (Figure 2). Similarly, it is a well known fact that tessellation of a sphere by uniform subdivision of spherical coordinates results in a surface with highly varying size of surface triangles. Alternative solutions

(Baumgardner and Frederickson, 1985) are often used when working with sphere approximations, such as when drawing a 3D sphere approximated by polygons.

Edelsbrunner and Mücke introduced the concept of alpha shapes (Edelsbrunner and Mücke, 1994) as a solution to find a geometric representation of the volume and surface of a set of points. An alpha shape is constructed by first computing the 3D Delaunay triangulation of the input points. Then, a α-parameter is used to decide which tetrahedra, triangles, edges and points should remain a part of the shape. Varying the parameter, shapes from the convex hull to the set of individual points can be constructed. Cholewo and Love utilized alpha shapes to find the gamut of devices and images (Cholewo and Love, 1999). They suggested that, for this application, the optimal gamut could be found by interactively changing the parameter value.

Giesen et al. (Giesen et al., 2005) proposed that what they called a discrete flow complex could be used to compute image gamuts. The colour space is divided into a grid, and for each grid point the distance to the nearest sample in the point data is calculated. This discrete distance map defines a flow direction for all positions in the colour space, and by traversing this structure the gamut of the image is determined.

2. Motivation

We have previously (Bakke et al., 2006) evaluated the performance of several gamut boundary determination algorithms on simulated device data. The Balasubramanian and Dalal modified convex hull algorithm was proven to perform well, particularly when 0.2 is used as the γ parameter value. However, the algorithm may result in a very large number of surface points, some of which form very small surface triangles. By combining existing methods for gamut determination with a more uniform segmentation algorithm, a good approximation of the gamut boundary can be found that uses a smaller number of surface points. The reduced surface complexity makes it possible to use less data to describe the gamut, which can be a factor when embedding gamut information in profiles or other files suitable for transfer and storage.

Perhaps more importantly, calculations of intersection points between the gamut surface and lines can be done in less time, improving the performance of gamut mapping algorithms. The segment maxima method provides a more compact gamut description, but its performance strongly depends on selecting a correct number of segments (Bakke et al., 2006), and generally gives a less accurate gamut boundary.

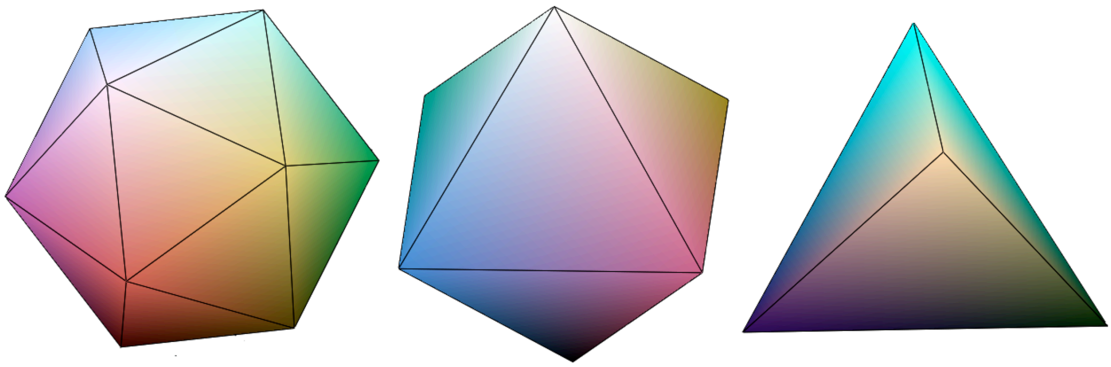

Figure 3. The geometric objects used as a basis for sphere tessellation: Icosaheron, octahedron and tetrahedron.

3. Method

Our method computes the gamut boundary from any arbitrary selection of data points in CIELAB or a similarly structured colour space, given that a sufficiently dense sampling of the entire gamut volume has been performed. First, a triangle-based approximation of a sphere is computed using a well-known tessellation technique. A basic triangle-tessellated geometric object is generated as an initial sphere approximation, using a tetrahedron, octahedron or icosahedron (Figure 3). A subdivision of the surface triangles is performed as shown in Figure 4. Each triangle is subdivided into 4 smaller triangles recursively, until the desired number of triangles is achieved. The resulting vertices are then projected onto the surface of the sphere, giving a close approximation of a sphere with a more uniform triangle size than simple subdivision of spherical coordinates as used in segment maxima.

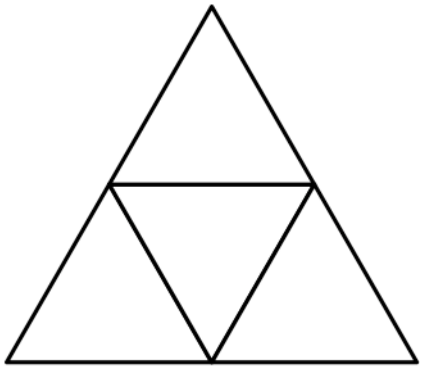

Figure 4. The subdivision of a triangle

Upon construction of the tessellated sphere, a data structure is generated that includes the neighbourhood connectivity information for the triangles. Each surface triangle defines a segment around the centre of the colour space (alternatively the centre of the gamut), represented by the tetrahedron given by the 3 triangle vertices and the centre point. Figure 5 shows a surface triangle as solid lines, while the tetrahedron defined by the triangle and the centre point is illustrated using dashed lines. Each pair of vertices P_0, P_1, and P_2 defines an edge of the triangle. The tetrahedron is surrounded by three other tetrahedra, each sharing a different edge of the triangle.

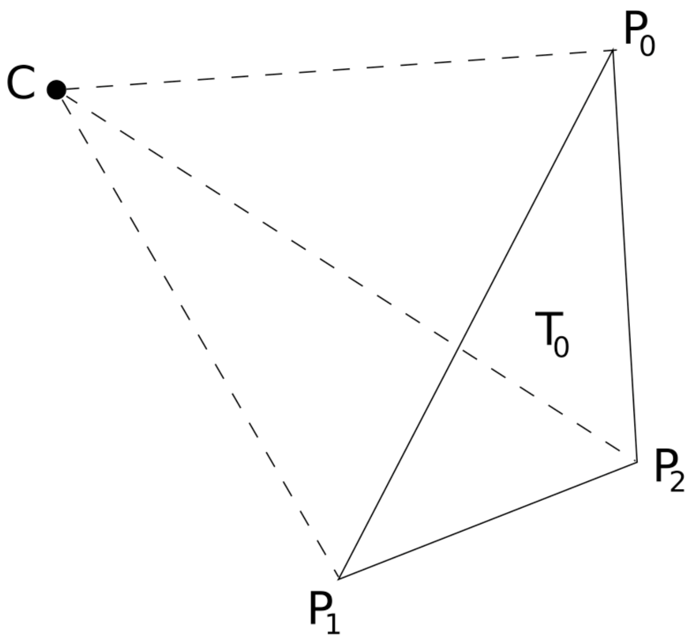

Figure 5. The surface triangle T_0 and the centre point C. The centre point and each pair of the triangle vertices define planes used to check whether a point is inside the segment.

We then process the data points that belong to the gamut by finding the enclosing segment for each point. In order to determine which segment encloses each data point, a simple algorithm has been implemented. Start with an arbitrarily chosen segment, and evaluate the plane equation for the 3 planes defined by the centre point and each of the 3 edges of the segment surface triangle for the data point position. If any of these computations show that the point is on the outside of the segment, use the neighbouring segment on the other side of that plane as the basis for another iteration of this search. Since consecutive data points usually are in the same area of the colour space, especially for images, the segment that contains the previous data point should be used as the starting segment for the next point. This ensures that the number of segments that must be traversed to identify the location of the data points is low.

When the enclosing segment has been located for each point, the length of the radius from the centre point to the data point is calculated. Similarly to the segment maxima method, only the single maxima point per segment that has the largest radius from the centre point is kept for the next parts of the algorithm.

When all the input points have been processed, it is quite common to find that some of the segments do not contain any data points. The original segment maxima method solves this by creating artificial points from interpolation of surrounding maxima points. However, this interpolation phase can be quite slow when the number of segments is high. To avoid this operation, we suggest the use of the modified convex hull algorithm with $\gamma = 0.2$ on the maxima points from the non-empty segments to create the final surface triangles that constitute the gamut boundary.

4. Results and discussion

We have utilized the method previously described in (Bakke et al., 2006) to compare this gamut boundary determination method against the often-used segment maxima method. By establishing a reference gamut boundary using a device model, gamut boundaries found using these methods on different data sets from the device can be compared with the reference to compute a relative volume mismatch using a voxel technique. Results based on data from a total of 32 different data sets from 2 device gamuts are plotted in Figure 6, where the average relative gamut mismatch of different segment numbers is shown. Our method performs well, especially as the number of segments increases. As the number of segments increases, the number of points preserved to the final surface calculation approaches the total number of points. Similarly, the performance approaches that of the Balasubramanian and Dalal modified convex hull used in the final stage of the method.

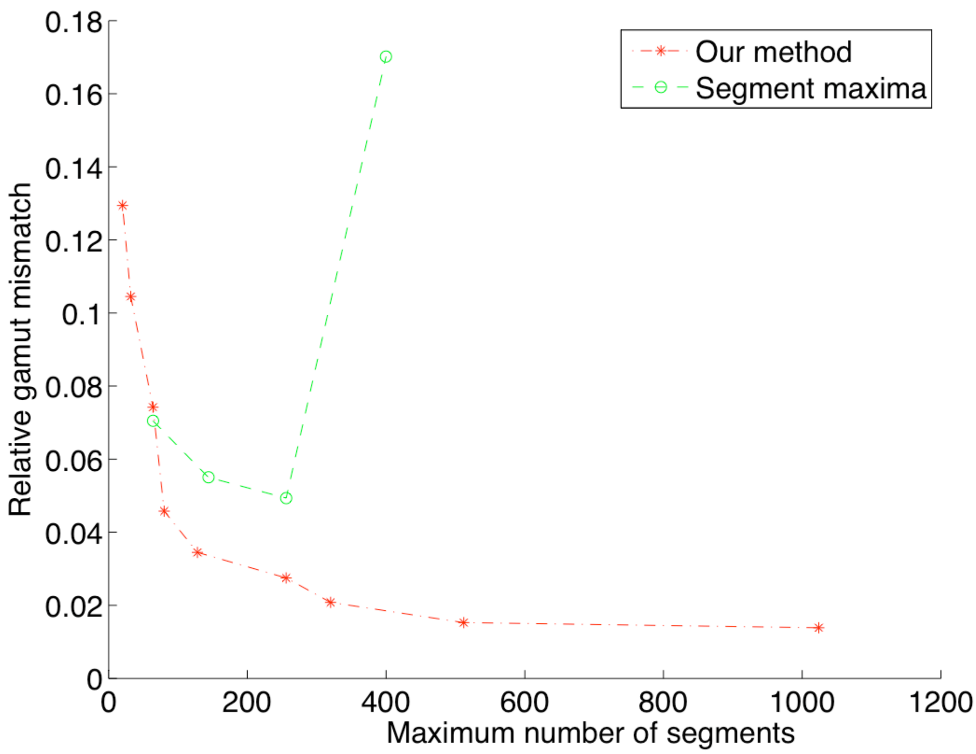

Figure 6. The average relative gamut mismatch of our method compared with segment maxima.

The segment maxima method performs poorly with a high number of segments on generic data sets because internal points end up being added to the gamut surface. Due to the computational complexity of the interpolation method used by the reference segment maxima implementation when constructing points for empty segments, we did not use the segment maxima method with more than 400 segments. An example of the gamut boundaries computed using our method can be seen in Figure 7, based on the same input points as the previous two gamut boundaries in Figure 1 and Figure 2.

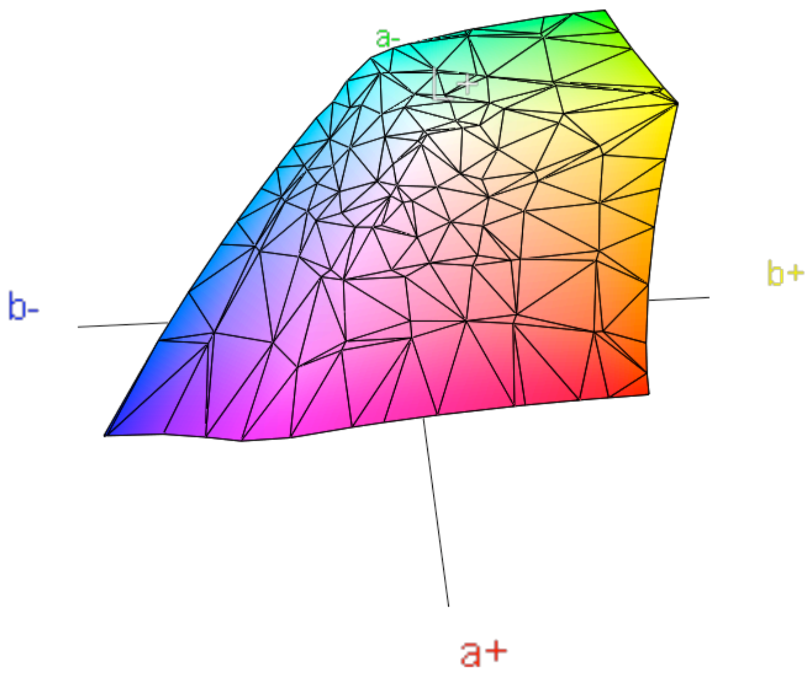

Figure 7. A gamut surface created using our method

5. Conclusions and future work

Our method can be used to find gamut boundaries with a specified maximum number of boundary points. We have compared the performance of this algorithm with the performance of segment maxima, and found that it is less sensitive to the number of segments used to approximate the gamut. This method leads to a reduced number of triangles compared to modified convex hull, making gamut intersection calculations faster. At the same time the performance approaches that of the modified convex hull.

We have looked at a method for reducing the number of input points used to generate the gamut surface. Alternatively, various mesh simplification algorithms (Cignoni et al., 1998) can be used to reduce the number of triangles in an existing surface structure. An interesting continuation of this work would be to compare our method to a surface generated using a combination of, e.g., modified convex hull, and a mesh simplification algorithm suitable for colour gamut surfaces.

6. Literature

Bakke, A. M.; Hardeberg, J. Y.; and Farup, I. (2006). "Evaluation of Gamut Boundary Descriptors". In Proceedings of IS&T and SID's Fourteenth Color Imaging Conference, pages 50–55, Scottsdale, Arizona.

Balasubramanian, R. and Dalal, E. (1997). "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.

Barber, C. B.; Dobkin, D. P.; and Huhdanpaa, H. (1996). "The Quickhull Algorithm for Convex Hulls". ACM Transactions on Mathematical Software, 22(4), pp. 469–483.

Baumgardner, J. R. and Frederickson, P. O. (1985). "Icosahedral Discretization of the Two-Sphere". SIAM Journal on Numerical Analysis, 22(6), pp. 1107–1115.

Bolte, S. B. (1992). "A perspective on non-impact printing in color". In Color Hardcopy and Graphic Arts, volume

1670 of Proc. SPIE, pages 1–11.

Braun, G. J. and Fairchild, M. D. (1997). ”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.

Cholewo, T. J. and Love, S. (1999). ”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.

Cignoni, P.; Montani, C.; and Scopigno, R. (1998). ”A comparison of mesh simplification algorithms”. Computers & Graphics, 22, pp. 37–54(18).

Edelsbrunner, H. and Mücke, E. P. (1994). ”Three-Dimensional Alpha Shapes”. ACM Transactions on Graphics, 13(1), pp. 43–72.

Engeldrum, P. G. (1986). ”Computing color Gamuts of Ink-jet Printing Systems”. In SID Proceedings, volume 27, pages 25–30.

Giesen, J.; Schubert, E.; Simon, K.; and Zolliker, P. (2005). ”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.

Guyler, K. (2001). ”Visualization of Expanded Printing Gamuts Using 3-Dimensional Convex Hulls”. American Ink Maker, 79(9), pp. 36–56.

Guyler, K. (2007). ”Effects of Stochastic Screening on the Gamuts of Wide Gamut Printing Systems”. www.iscc.org/ISCC2007/abstracts/T15IG1_Guylar.pdf.

Hardeberg, J. Y. (2001). Acquisition and Reproduction of Color Images: Colorimetric and Multispectral Approaches. Dissertation.com, Parkland, Florida, USA.

Hardeberg, J. Y. and Schmitt, F. (1997). ”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. Also in R. Buckley, ed., Recent Progress in Color Management and Communications, IS&T, pages 88-91, 1998.

Herzog, P. G. (1996). ”Analytical Color Gamut Representations”. Journal of Imaging Science and Technology, 40, pp. 516–521.

Herzog, P. G. (1998). ”Specifying and visualizing colour gamut boundaries”. In MacDonald, L. W. and Luo, M. R., editors, Colour Imaging: Vision and Technology. John Wiley & Sons Ltd.

Inui, M. (1993). ”Fast Algorithm for Computing Color Gamuts”. Color Research and Applications, 18(5), pp. 341–348.

MacAdam, D. L. (1935). ”Maximum visual efficiency of colored materials”. J. Opt. Soc. Amer., 25, pp. 361–367.

Mahy, M. (1997). ”Calculation of Color Gamuts Based on the Neugebauer Model”. Color Research and Application, 22, pp. 365–374.

Meyer, G.W.; Peting, L. S.; and Rakoczi, F. (1993). ”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.

Meyer, G. W. and Robertson, C. A. (1997). ”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.

Morovic, J. and Luo, M. R. (1997). ”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.

Neugebauer, H. E. J. (1937). ”Die theoretischen Grundlagen des Mehrfarbenbuchdrucks”. Zeitschrift f¨ur wissenschaftliche Photographie, Photophysik und Photochemie, 36(4), pp. 73–89.

Reel, R. L. and Penrod, M. A. (1999). ”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.

Saito, R. and Kotera, H. (2000). ”Extraction of Image Gamut Surface and Calculations of its Volume.”. In Color Imaging Conference, pages 330–334.

Saito, R. and Kotera, H. (2004). ”Imagedependent three-dimensional gamut mapping using gamut boundary descriptor”. Journal of Electronic Imaging, 13(3), pp. 630–638.

Stone, M. C.; Cowan, W. B.; and Beatty, J. C. (1988). ”Color gamut mapping and the printing of digital color images”. ACM Transactions on Graphics, 7(4), pp. 249–292.

Viggiano, J. A. S. and Hoagland, W. J. (1998). ”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.

Zhao, X. (2001). ”Implementing an ICC Printer Profile Visualization Software”. Master’s thesis, School of Printing Manamement and Sciences in the College of Imaging Arts and Sciences of the Rochester Institute of Technology.