A New Device Color Gamut Boundary Description Algorithm Based on Irregular Segmentation

Zhang Yin, Jiang Yongwen, Lin Maohai*
School of Print and Package Engineering, Qilu University of Technology, Jinan, Shandong, China

Received 5 March 2015; revised 22 March 2016; accepted 23 March 2016

Abstract: The description of device color gamut is very important for color gamut mapping, which plays an important role in cross-media color reproduction. To get a more accurate color gamut boundary, several gamut boundary description (GBD) methods have been presented in the past; however, many problems still exist in the current algorithm (e.g., accuracy and speed). In this study, we propose a new device GBD method based on irregular segmentation and evaluate it against other GBD algorithms. In the irregular segmentation method, the device color gamut is divided into the high-chroma and the low-chroma color parts, and the high-chroma parts are divided into more segments through the CIELAB $a^*b^*$ plane when calculating color gamut descriptors. After that, in each segment, the radii between the color points and the center point are calculated. For the color points located in the high-chroma parts, the color point with the biggest radius is selected as the gamut boundary descriptor, and for that in the low-chroma parts, if the corresponding outer segments are empty, the color point with the biggest radius is selected as the gamut boundary descriptor. Then, the gamut boundary descriptors are visualized in CIELAB color space using triangulation method. Finally, the irregular segmentation method is evaluated in contrast with other GBD methods using gamut mismatch index. The results show that the irregular segmentation method performs better in accuracy and has smaller variance.

INTRODUCTION

Color gamut is the entire range of colors that can be rendered by a device or that are contained in an image.1–3 Device color gamut gives an intuitive view of what can be reproduced by a certain media, and the overlap of two visualized color gamuts can be used to show improvements or advantages of one printer over another.4 The overlap of an image color gamut and device color gamut is one of the factors needed to be considered when we choose the proper rendering intent which is very important in cross-media color reproduction, and this can also be used in image-dependent color gamut mapping.5 The color gamut can be constructed by gamut boundary descriptors. The construction of a gamut boundary is also the first step in the process of performing color gamut mapping,6 color gamut extension,7 spatial color gamut mapping,8,9 dependent primary control system,10,11 and so forth. With the visualized color gamut, it can be easier to analyze the reproduction effect of a certain printing material.12 The performance of a color GBD algorithm is very important in color gamut mapping.

MacAdam13 first attempted to get the device color gamut in 2D $xy$ color space. The color gamut boundary in 2D $xy$ chromaticity can be acquired by plotting the outline of the color which a device can render14; however, the 2D gamut lacks lightness information, which brings some limitations in the use of color gamut visualization and other applications to some extent. Thus, the 3D color gamut boundary offers a more clear and more intuitive representation of the color distribution of an image.15 To get an exact device color gamut, several GBD algorithms have been presented in the past.

GBDs Based on Convex Hull Algorithm

The convex hull of a set of points is the smallest convex polyhedron that contains all the points. The convex hull can be constructed by the convex combination from Eq. (1), in which $k$ is the number of the polyhedron’s
vertices, \( w_j \) is the non-negative weight which sums to one, \( p_j \) is the polyhedron’s vertices vector. The convex hull can be realized by quickhull algorithm. 16

\[
c = \sum_{j=1}^{k} w_j p_j
\]

By using the convex hull, we can get an approximation color gamut which contains all the colors. The color gamut constructed by convex hull algorithm always has larger volume and also has no concave parts in the constructed gamut surface4, 17, 18; however, there are often gamut concavities where convex hull algorithm fails.

The evolution of convex hull algorithm is the modified convex hull algorithm represented by Balasubramanian and Dalal. 19 In their method, the data are processed by a gamma function using Eq. (2) in which the new vertex coordinate \( \tilde{p} \) is calculated in the gamma function based on the central vertex \( \tilde{c} \) and parameter \( \gamma \) before the convex hull algorithm is applied. The results show that with the proper \( \gamma \), the modified convex hull algorithm can get an accurate gamut with concaves. However, it is difficult to decide the proper \( \gamma \), and the accuracy of the constructed gamut is also affected by the center point.

\[
\tilde{p}^j = [\tilde{p} - \tilde{c}]^{-1} (\tilde{p} - \tilde{c}) + \tilde{c}
\]

Another evolution of modified convex hull is presented by Bakke and Farup, 20 who used modified convex hull to calculate an accurate gamut boundary and then simplifies the gamut boundary using an established mesh decimation technique, and who claims that this method is also very accurate.

Even though the convex hull-based GBD algorithms can get an accurate color gamut, its descriptors are not equally distributed among the color gamut surface, which makes it time consuming to find line gamut boundary conducting color gamut mapping in constant hue plane.

The Alpha Shape Algorithm

The use of alpha shape to construct color gamut boundary is first proposed by Cholewo and Love. 21 The gamut is constructed by a subset of Delaunay tessellations of color points which a device can render.

The biggest problem in using alpha shape algorithm is that it is difficult to decide the proper \( \alpha \), although Cazals and coworkers 22 tried to use the dynamic determination of \( \alpha \) to improve the performance.

GBDs Based on Segment Maximum Algorithm

The segment maximum algorithm is presented by Morovic and Luo. 23, 24 The color space is divided by hue angle \( \alpha \) and lightness angle \( \theta \) into \( m \)-by-\( n \) segments, storing the color point with the maximum radius in each segment as the gamut boundary descriptor. To get the color gamut along arbitrary line of constant hue angle, Morovic also developed the flexible sequential line gamut boundary (FSLGB) method 23, 26, 27 which makes it easier to be used in color gamut mapping.

In the segment maximum-based methods, the calculation error of the gamut boundary descriptors located in high-chroma parts is larger than that located in the low-chroma parts; it always gets a nonuniform gamut boundary where there are more effective gamut boundary descriptors located in the two ends of the color gamut [i.e., the highlights (high \( L^* \) values) and the shadow area (low \( L^* \) values)] than those located in high-chroma parts in the central area.

An evolution algorithm is presented by Bakke et al., 28 who used modified convex hull to construct the color gamut. Bakke claimed that his method performed better than the segment maximum algorithm in terms of accuracy, particularly for higher number of segments. However, this method still cannot avoid the nonuniformity of effective gamut boundary descriptors located in the high-chroma parts in the middle of the lightness axis and the low-chroma parts in the two ends of the color gamut.

Even though the segment maximum algorithm methods still have some problems, they can be easily used in color gamut mapping algorithms and other applications in color management. It is also used in Public Gamut Mapping Algorithms released by the CIE, 29 which can be used to construct proper color tables in ICC profiles 30; the segment maximum algorithm is also used in little CMS, which is a popular open-source color management engine. 31

THE IRREGULAR SEGMENTATION GBD METHOD

In high-quality color reproduction, the high-chroma color that is located in the outer part of the color space is the most important color to reproduce, and it is difficult to determine the color gamut boundary of those colors accurately because of the segmentation method leading to nonuniform gamut boundary descriptors. What is more, in color gamut mapping, the out-of-gamut determination mainly deals with those colors as well. For the low-chroma colors, they are usually located in the inner part of the color space, most of which are included inside the color gamut, and it is less important in color gamut boundary determination or the out-of-gamut determination in color gamut mapping.

However, the segment-based methods in gamut boundary determination (e.g., the segment maximum GBD), they treat the gamut boundary descriptors located in the outer parts and those located in the inner parts in the same way. This leads to the results that the segments are denser in the two ends and sparser in the important middle parts of the gamut, as in general, the color gamut is larger in the middle parts and smaller in the two ends along the \( L^* \) axis in geometry shape. In this way, the effective gamut boundary descriptors located in the
middle high-chroma parts would be less than those located in the low-chroma parts in the two ends of the color gamut. 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.

As there is only one effective gamut boundary descriptor in each segment, even though interpolation algorithm is used in the later steps, the accuracy would not be improved much in substance. Therefore, the accuracy of gamut boundary in each segment can be expressed by the length of arc which is observed across the gamut boundary descriptor as shown in Fig. 1; the accuracy of segment OAB can be expressed by \(\text{ArcAEB} = \pi d_1\), and the accuracy of segment OCD can be expressed by \(\text{ArcCFD} = \pi d_2\), where \(\alpha'\) is the central angle and \(d\) is the radius corresponding to the segment. The accuracy rate between the two parts is \(d_1 = d_2\), as \(d_1 > d_2\) that is, if a gamut boundary descriptor is located in the inner part such as \(F\), it will be more accurate than the color gamut boundary descriptor located in the outer part such as \(E\) in describing color gamut.

The inaccuracy problem in segment-based GBDs stems from the algorithm itself. Figure 2 shows the first area of a rough device gamut boundary constructed by segment maximum algorithm after interpolation on \(a^*b^*\) plane in CIELAB color space. In this area, the color space is divided into four segments; in the gamut boundary determination, color \(H\), \(I\), \(G\), and \(K\) are chosen as the gamut boundary descriptor of segments 1, 2, 3, and 4. This leads to the colors in area 5, 6, 7, and 8, which located in the real gamut would be determined as the out-of-gamut colors as their distances to the center point are less than those of the gamut boundary descriptors, and this problem would be more obvious as the distance increases. However, these are the blind areas in the segment-based gamut boundary determination algorithms. This determination mistake cannot be avoid in segment-based methods, and the only way to reduce this kind of mistake is to increase the number of segments, to one extreme, if the color space is divided into an infinite number of segments; this problem will not exist anymore. However, with the increase of the segments, the segments would become more denser in the two ends of the color gamut, and more empty segments will occur. Even though those empty segments can be filled by interpolation gamut boundary descriptors, the accuracy of the gamut boundary would not be improved much, it would consume much more time and memory space.

Based on this phenomenon, a new gamut boundary determination method is needed to improve the accuracy of the gamut boundary descriptors located in high-chroma color parts and also not to increase the empty segments in low-chroma color parts in the two ends of color gamut. In this article, we present an irregular segmentation method to solve this problem.

**Irregular Segmentation Model**

In the irregular segmentation model, the device gamut is divided into a high-chroma part and a low-chroma part on \(a^*b^*\) plane; along the \(L^*\) axis, the gamut is divided into several layers as shown in Figs. 3 and 4.

On \(a^*b^*\) plane, the device gamut is divided into two parts by \(b\) (or more if necessary, e.g., another parameter \(c\) is used to divide the color space into the outer, middle, and inner parts, which would result in better accuracy and also cost more time). In the inner part, the segments are divided by an angle parameter \(\gamma\) which is in degree, and in the outer part, the segments are divided by \(\gamma/n\) \((n\) is the amount of segments in the outer part corresponding to the inner part); thus, \(360/\gamma\) segments and \(360n/\gamma\) segments are included in the inner part and outer part on \(a^*b^*\) plane, respectively. In this way, the segments in the outer part are \(n\) times more than the inner part. Along the \(L^*\) axis, the gamut is divided by parameter \(h\) \((h\) is the lightness distance of each segment along the \(L^*\) axis), and then the device gamut is divided into \([360(1+n)/\gamma](100/h + 1)\) segments totally.
After performing the segmentation operation, the accuracy rate in Fig. 1 becomes \( d_1 = nd_2 \). For the gamut boundary descriptors located in high-chroma parts, the accuracy is \( n \) times smaller than the traditional method; this guarantees a more uniform result between the gamut boundary descriptors located in high-chroma parts and those located in low-chroma parts.

The Irregular Segmentation Method

To calculate the gamut boundary of the device, the following steps are used.

1. An \([360(1+n)/x](100/h+1)\) empty matrix Gamut is set to store the gamut boundary descriptors and to use the index of the matrix to locate the segments between the outer parts and the inner parts. Note that in this study, the gamut boundary descriptors are stored in one-by-one layer from low to high \( L^* \) values, and in each layer, low-chroma values are stored followed by high-chroma values.

2. Calculate the radius \( r \), the hue angle \( \theta \), and the layer \( f \) of each color using Eq. (3).

\[
\begin{align*}
\theta &= \tan^{-1}(b/a) \\
r &= \sqrt{a^2 + b^2} \\
f &= \frac{L}{h}
\end{align*}
\]

where \( \theta \) is the hue angle of each color, \( \tan^{-1}(b/a) \) is the inverse of the tangent trigonometric function, and the \( \theta \) needs to be transformed to the range \([0^\circ, 360^\circ]\). \( r \) is the distance between the color and the corresponding center point in \( L^* \) axis on the same \( a^*b^* \) plane, which is the equivalent of chroma value of a color, \( f \) is the layer of the segment in vertical direction of CIELAB color space.

3. Calculate the index of the gamut boundary descriptor matrix for a given color gamut boundary descriptor.

4. Determine the gamut boundary descriptor of each segment.

To divide the device gamut into the high-chroma part and the low-chroma part, parameter \( \beta \) is used. Use the coordinate in step 2, if \( r > \beta \), which means that the color gamut boundary descriptor is located in the high-chroma part, and then use Eq. (4) to calculate the gamut boundary descriptor index. If \( r < \beta \), which means that the color gamut boundary descriptor is located in the low-chroma part, the gamut boundary descriptor index is calculated by Eq. (5).

\[
\begin{align*}
m &= \left[360/\alpha + (\beta/h) \times 360(1+n)/\alpha \right] \\
m &= \left[360/\alpha + (\beta/h) \times 360(1+n)/\alpha \right]
\end{align*}
\]

4. Determine the gamut boundary descriptor of each segment.

For each member in matrix Gamut, use the matrix index to determine whether the segment is located in the high-chroma part or in the low-chroma part by Eq. (6). If \( k \) is larger than \( 360/\alpha \), then the corresponding segment is located in the high-chroma part or it is in the low-chroma part. In the first case, use color point with the maximum radius as the gamut boundary descriptor. In the latter case, use Eq. (7) to calculate the corresponding high-chroma segments and if Gamut\(_{m1}\), Gamut\(_{m2}\), Gamut\(_{m3}\), ... Gamut\(_{mn}\) are all empty, then use color point with the maximum radius in the low-chroma part as the gamut boundary descriptor or else use the color point with the maximum radius in segment \( m_1, m_2, m_3, \ldots m_n \) as the gamut boundary descriptor.

\[
k = m - \left[360(1+n)/\alpha \right] \times \left[360(1+n)/\alpha \right]
\]
After this process, all the device gamut boundary descriptors are recorded in matrix $Gamut$, as in this method, the gamut is divided into two parts, the gamut boundary descriptors should only exist in inner parts or the corresponding outer parts, and if they are both empty, there are no gamut descriptors in these segments. If there are many empty segments in both the inner part and the corresponding outer parts at the same time, increase the values $a$ and $h$ or use interpolation algorithm between the neighboring segments. Then, the device color gamut descriptors are visualized in the color space using a triangulation algorithm. Figure 5 shows the color gamut of Adobe1998 using irregular segmentation method.

$$
m_1 = \frac{360}{x + nk}
$$
$$
m_2 = \frac{360}{x + nk - 1}
$$
$$
m_3 = \frac{360}{x + nk - 2}
$$
$$
\ldots
$$
$$
m_n = \frac{360}{x + nk - (n-1)}
$$

EVALUATION AND RESULTS

Evaluation

The GBD algorithm is very important in color gamut mapping, and it is also necessary to know the accuracy of each GBD algorithm. However, there is relatively little research about the GBD’s evaluation. CIE TC8-05 defined a set of metrics for evaluating gamuts, and these metrics can be used to make some comparisons between gamuts. However, they do not give information on how well one gamut can match another in terms of both intersection and volume. In the methods proposed by Braun and Fairchild$^{23}$ and Green and coworkers,$^{34}$ the GBD algorithms are evaluated by calculating the color difference between GBDs and some known gamut boundary descriptors, although it is difficult to find enough known gamut boundary descriptors. Morovic$^{26}$ also pointed out that the gamut volume, gamut intersection, and gamut surface smoothness can be used in GBD evaluation. In the method proposed by Bakke et al.$^6$ the different GBD algorithms are evaluated by gamut mismatch index; however, the mismatch is not perfect, for example, it does not work well under condition $c$ in Fig. 6 due to the lack of consideration of the constructed gamut volume. In the method proposed by Deshpande et al.$^{35}$ gamut similarity is evaluated by gamut comparison index (GCI); however, GCI cannot quantify the real difference between two gamuts, and it is impossible to know whether one gamut is overestimated or underestimated by another gamut.

Fig. 5. The color gamut of Adobe1998 using irregular segmentation method.

Fig. 6. The spatial relationship between the projections of constructed gamut and reference gamut on CIELAB a*b* plane.
which was constructed with a different GBD algorithm. In their evaluation method, the reference gamut is constructed by convex hull or alpha shape algorithm, which makes it impossible to be used in the evaluation of different GBD algorithms, because this would not be fair for other GBD algorithms.

Since the constructed gamut and the real device gamut are always in irregular shape, and the constructed gamut is not simply overestimated or underestimated, the real gamut as is shown in Fig. 6, there are also many different gamuts that have the same volume. If the intersection gamut is much smaller than the constructed gamut, it should not be considered as a well-constructed gamut. Thus, it is not enough just to consider the volume and the intersection volume. In the evaluation, the constructed gamut, reference gamut, and intersection gamut should all be considered.

In this study, different GBDs are evaluated by a new gamut mismatch index, and to have an in-depth view of the performance of different GBDs, the high-chroma parts and the low-chroma parts are also evaluated.

To evaluate different GBD methods, the reference gamut for different devices should be constructed first. The reference gamut should contain all the colors that can be reproduced by the device exactly, and it should not be based on a certain GBD algorithm. As the effective color gamut of the device encoding is the result of transforming the device encoding into a colorimetric description for specific viewing conditions (called the PCS) using colorimetric rendering intent. With the colorimetric rendering intent, all the device colors that lie outside the effective gamut can be mapped to in-gamut PCS value; this method is used to simulate the output process of a device, and even though the ICC profile has its own gamut tag or GBD methods, it would not bring any effects on the data processing in the next step. The reference gamut can be constructed by performing a conversion from dense samples of the CMYK/RGB color space to CIELAB color space using device ICC profile with colorimetric rendering intent, and then, those colors are quantized by voxel processing method using Eq. (8), where \( k \) is the side length of the voxel, \( x' \) is the color to be processed, and \( x \) is the color after voxel processing (all the variables are in CIELAB units). The resulting gamut would contain all the possible colors which can be reproduced by that device. Another advantage to use voxel gamut as reference gamut is that it is not a model-based gamut (e.g., the modified convex hull reference gamut), and it is fair to all the GBD algorithms in the evaluation.

\[
x = \begin{cases} 
(x'-1)/k \times k + \frac{k}{2} & x' \geq 0 \\
[(x'-1)/k] \times k + \frac{k}{2} & x' < 0
\end{cases}
\]

In this experiment, we use \( r_{mismatch} \) in Eq. (10) as the gamut mismatch index. In Eq. (9), \( V \) is the gamut volume constructed by the GBD algorithms, \( V_{ref} \) is the reference gamut volume, and \( V_i \) is the intersection gamut volume between the reference gamut and the gamut constructed by the GBD algorithm. While using this metric, \( r_{mismatch} = 0 \) means that the constructed gamut is the same as the reference gamut, \( r_{mismatch} > 0 \) means that the reference gamut is overestimated, \( r_{mismatch} < 0 \) means that the reference gamut is underestimated, and the more \( r_{mismatch} \) is close to 0, the better is the accuracy of the GBD algorithm. The gamut mismatch index can also be used in evaluating the similarity of different gamuts.

\[
r_{mismatch} = \begin{cases} 
\frac{|V - V_i| + |V - V_{ref}|}{V_{ref}} & V \neq V_{ref} \\
\frac{|V - V_i| + |V - V_{ref}|}{V_{ref}} & V = V_{ref}
\end{cases}
\]

To evaluate the new algorithm’s performance in contrast with other GBDs in high-chroma and low-chroma parts, we also use Eq. (10) to evaluate the accuracy of gamut boundary descriptors located in the high-chroma parts and low-chroma parts.

### The Calculation of Volume and Volume Intersection

As the gamut surfaces of convex hull, modified convex hull, USV, and irregular segmentation algorithms are constructed by triangulation method, the volume of the gamut can be calculated by Eq. (10), where \( k \) is the number of surface triangles included in the triangulation profile, \((x_i, y_i, z_i)\) is the vertex of a surface triangle, and \((x_0, y_0, z_0)\) is the coordinate of the central point of color space.

\[
V = \sum_{i=1}^{k} \frac{1}{6} \begin{vmatrix} 
1 & 1 & 1 \\
x_1 & x_2 & x_3 \\
y_1 & y_2 & y_3 \\
z_1 & z_2 & z_3 
\end{vmatrix} 
\]

In the tetrahedron, which contains a surface triangle and the central point, Eq. (11) is used to calculate the volume of the graphic which is composed by the tetrahedron and any point in the color space, where \((x_k, y_k, z_k)\) is the coordinate of any point in the color space and \((x_0, y_0, z_0)\) is the vertex coordinate of tetrahedron. If \( V_{tetrahedron} \), it turns out that the point is located in the inner part of the tetrahedron or it is located in the outer part of the color space. This method can be used to determine whether a point is located in the tetrahedron. The uniform voxels in the CIELAB color space contained by the gamut can be determined by this method. The gamut intersection volume can be acquired by intersection operation between the voxels contained in the reference gamut and those contained in the constructed gamut.
TABLE I. The profile used in the experiment.

<table>
<thead>
<tr>
<th>No.</th>
<th>Device profile</th>
<th>Profile type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>USWebcoated</td>
<td>Output</td>
</tr>
<tr>
<td>2</td>
<td>Japancolor2001coated</td>
<td>Output</td>
</tr>
<tr>
<td>3</td>
<td>UncoatedFOGRA29</td>
<td>Output</td>
</tr>
<tr>
<td>4</td>
<td>Canon ipf 8300</td>
<td>Output</td>
</tr>
<tr>
<td>5</td>
<td>Epson 7800pro</td>
<td>Output</td>
</tr>
<tr>
<td>6</td>
<td>Adobe RGB(1998)</td>
<td>Generic</td>
</tr>
<tr>
<td>7</td>
<td>AppleRGB</td>
<td>Generic</td>
</tr>
<tr>
<td>8</td>
<td>sRGB</td>
<td>Generic</td>
</tr>
<tr>
<td>9</td>
<td>Colormatch RGB</td>
<td>Generic</td>
</tr>
<tr>
<td>10</td>
<td>Widegamut RGB</td>
<td>Generic</td>
</tr>
<tr>
<td>11</td>
<td>Fuji Frontier scanner</td>
<td>Scanner</td>
</tr>
<tr>
<td>12</td>
<td>Kodak Professional HR 500</td>
<td>Scanner</td>
</tr>
<tr>
<td>13</td>
<td>Apple Cinema</td>
<td>Display</td>
</tr>
<tr>
<td>14</td>
<td>EizoColorEdge CG246</td>
<td>Display</td>
</tr>
<tr>
<td>15</td>
<td>EizoColorEdge CG276</td>
<td>Display</td>
</tr>
<tr>
<td>16</td>
<td>EizoColorEdge CG303w</td>
<td>Display</td>
</tr>
<tr>
<td>17</td>
<td>NEC MultiSync PA271W</td>
<td>Display</td>
</tr>
</tbody>
</table>

\[ V' = \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_k & x_2 & x_3 & x_0 \\ y_k & y_2 & y_3 & y_0 \\ z_k & z_2 & z_3 & z_0 \end{vmatrix} + \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_k & x_2 & x_3 & x_0 \\ y_k & y_2 & y_3 & y_0 \\ z_k & z_2 & z_3 & z_0 \end{vmatrix} + \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_k & x_2 & x_3 & x_0 \\ y_k & y_2 & y_3 & y_0 \\ z_k & z_2 & z_3 & z_0 \end{vmatrix} \]  

(11)

**Experiment and Results**

In this experiment, we use 17 profiles (five output device ICC profiles, five generic ICC profiles, five display ICC profiles, and two scanner ICC profiles) to evaluate different GBD algorithms. The profiles used in evaluation are given in Table I. To convert the dense samples (the samples are made every one unit in device color space) from device color space into CIELAB color space, a color management module (CMM) in Matlab\textsuperscript{32} is used with relative colorimetric rendering intent through the device ICC profiles. The voxels that each reference gamut contains are \(2 \times 2 \times 2\) in size in CIELAB color space; the maximum color difference of each voxel is 1.73. (The evaluation is done in Matlab\textsuperscript{32} and the CPU is intel\textsuperscript{®} Core\textsuperscript{TM} 2 Duo E8500 3.16 GHz.)

As the modified convex hull performs best in the convex hull-based GBD methods, in the segment-based methods, the USV performs best, and convex hull always contains every color that the device can reproduce; these three GBD algorithms are chosen to compare with irregular segmentation method. As is proved by Bakke et al.,\textsuperscript{6} with \(\gamma = 0.2\), the modified convex hull performs best. In the experiment, we also use \(\gamma = 0.2\) in modified convex hull.

---

**Fig. 7.** The gamut mismatch index of different GBD algorithms.

**Fig. 8.** The gamut mismatch index of high-chroma parts.

**Fig. 9.** Gamut mismatch index of low-chroma parts.
hull and 24 × 18 segments in USV. In irregular segment GBD, $\beta = 20$, $\alpha = 15^\circ$, and $h = 5$ are used which divide the color space into $24 \times 21$ in the low-chroma parts, and $72 \times 21$ in the high-chroma parts $(50, 0, 0)$ is selected as the center point in CIELAB color space calculating the gamut volume.

The evaluation results are shown in Figs. 7–13. Figures 7–9, 11, and 13 adopt boxplot which is a convenient way of graphically depicting groups of numerical data through their quartiles. Figure 7 shows that both the convex hull and the modified convex hull algorithm overestimate the reference gamut, and both USV and irregular segmentation algorithm underestimate the reference gamut. The modified convex hull and the irregular segmentation algorithm almost have the same accuracy; however, irregular segmentation algorithm has smaller variance. The accuracy of convex hull algorithm is the worst. Figure 8 shows that in the high-chroma parts, the accuracy of irregular segmentation method is the best, the accuracy of modified convex hull is better than the USV algorithm, and the convex hull algorithm is the worst. Figure 9 shows that in the low-chroma parts, the accuracy of USV and convex hull algorithm is much better than that in the high-chroma parts, and the accuracy of modified convex hull and irregular segmentation algorithm is almost the same as that in the high-chroma parts. Figure 10 shows the correlation coefficient between gamut mismatch index and the reference gamut volume. It reveals that the effect of reference gamuts’ volume on GBD algorithm evaluation. From Fig. 10, in the high-chroma parts, the correlation coefficients are almost the same as those in the whole gamut. The modified convex hull algorithm has a high correlation coefficient in the high-chroma parts and the whole gamut (about 0.84), this suggests that with the increase of reference gamut volume, the mismatch index also increases, which means that the GBD’s accuracy would decrease, which makes it not very suitable describing big gamut in the use of color gamut visualization. The reference gamut volume has less influence on the accuracy of irregular segmentation, convex hull, and USV. Figure 11 shows that the convex hull algorithm is less time consuming, as the time in modified convex hull is almost as same as the irregular segmentation algorithm and the USV is most time consuming. Figure 12 shows the volume of the 17 device gamuts. Figure 13 shows the calculation error of the voxel processing method when compared with geometry method in volume computation.

![Fig. 10. Correlation coefficients analysis.](image1)

![Fig. 12. The volume of the different device.](image2)

![Fig. 11. The consumed time of each GBD algorithm.](image3)

![Fig. 13. The calculation error of voxel process method.](image4)
as the calculation error is 0.78% on average, and the voxel processing method is very accurate.

The results from the experiment suggest that the irregular segmentation GBD method performs as accurate as modified convex hull, and it is less affected by the gamut volume of the reference gamut. As its gamut boundary descriptors are determined in segments and are distributed uniformly across the entire gamut, it is easier to be used in color gamut mapping and other applications in color management. The irregular segmentation algorithm also performs much better in the high-chroma parts and faster than the USV algorithm, and it has almost the same accuracy in the high-chroma and low-chroma parts. The performance of USV and convex hull algorithm is much different in the high-chroma and low-chroma parts.

CONCLUSIONS

We have introduced a new device GBD method based on irregular segmentation and evaluated it against the current presented GBD algorithms using gamut mismatch index. The results show that the irregular segmentation algorithm can perform very fast and also guarantee a good accuracy especially in the high-chroma parts. As its gamut boundary descriptors are determined in segments, this suggests that this algorithm also has an associated FSLGB algorithm, which would make it easier to be used in color gamut mapping, and it is also the next step of our research.

