Gamut mapping using variable anchor points

Chae-Soo Lee*, Kyeong-Man Kim**, Eung-Joo Lee***, and Yeong-Ho Ha*

*School of Electronic and Electrical Engineering,
Kyungpook National University, Taegu 702-701, Korea

**Printer Div., Samsung Electronics Co., Ltd.

*** Dept. of Information/Communications Engineering,
Tongmyong University of Information Technology

ABSTRACT

Recently, a variety of imaging devices are being used to represent electronic color images. The reproduced color, however, is different from the original color because of the difference of producible colors on the devices. The range of producible colors offered by a device is referred to as its gamut. In this paper, a gamut-mapping algorithm (GMA) is proposed that can maintain device-independent color. Categorized as a parametric GMA, this algorithm utilizes variable anchor points (a center of gravity on the lightness axis) to both reduce a sudden color change on the gamut boundary of the printer and to maintain a uniform color change during the mapping process. Accordingly, the proposed algorithm can reproduce high quality images with low-cost color devices.

Keywords: color matching, gamut mapping, anchor point, lightness, chroma

1. INTRODUCTION

Some practical output systems are only capable of producing a limited range of colors. The range of producible colors on a device is referred to as its gamut. Often, an image will contain colors that are outwith the gamut of the target output device. In such a case, the image colors must be mapped within the gamut, which requires the use of a gamut-mapping algorithm (GMA) [1-4]. Conventional GMAs can be divided into three groups; successive, simultaneous, and parametric GMAs [5]. A successive GMA maps the perceptual attributes (hue, saturation, and lightness) separately [6-8]. A simultaneous GMA maps the colors so that all of their attributes are changed simultaneously. A parametric GMA changes the color behavior based on either the shape of the original and reproduction gamuts at a particular hue angle or some other user-defined parameter [9,10].

Conventional parametric GMAs use an anchor point (a center of gravity on the lightness axis) to decrease the lightness of the bright part of an original image while increasing the lightness of the dark part. However, sudden color changes between the inside and outside of the gamut often result from this process. Accordingly, a parametric GMA that utilizes variable anchor points is, therefore, proposed as a solution for matching the colors produced on a monitor with those on a printer. The proposed GMA consists of lightness mapping and parametric mapping using variable anchor points with a constant slope. Lightness mapping linearly maps the lightness of the original image on the monitor into the lightness range of the printer. In addition, separate mapping methods are used to produce an approximately uniform color change for colors outwith the printer gamut. This separate mapping classifies regions according to the lightness of the monitor and printer gamuts. Based on these regions, the colors of the bright and dark regions in an input image are then mapped into the printer gamut by clipping their chroma component along the lines of a constant slope. In the case of a middle region, mapping is performed towards an anchor point. Consequently, separate mapping utilizing variable anchor points with a constant slope can reproduce a continuous tone image in the boundary region of a printer gamut and reduce mapping errors.

Part of the IS&T/SPIE Conference on Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts IV • San Jose, California • January 1999
SPIE Vol. 3648 • 0277-786X/98/$10.00

2. GAMUT MAPPING ALGORITHMS

2.1 SUCCESSIVE GMAS

The characteristic of this group is that each of the perceptual attributes is mapped separately, i.e. in most cases the algorithm has two stages: the mapping of lightness and the mapping of chroma. All the algorithms in this group (LLIN, LNLIN, LCLIP and LSLIN) first map lightness linearly so that the minima and maxima of the two gamuts are mapped onto each other. The LLIN [7] algorithm linearly compresses chroma along lines of constant lightness, the LNLIN [6] algorithm makes this compression in a non-linear way and the LCLIP [8] algorithm clips the out-of-gamut colors onto the gamut boundary and leaves in-gamut color unchanged. The LSLIN [7] algorithm is a hybrid algorithm as it maps colors towards a point on the L^* axis (L^*=50) after the initial lightness mapping, in which it changes lightness at two stages of the process (see Fig. 1). Note that all successive and simultaneous algorithms evaluated here preserve hue, use the image gamut as their source (original) gamut and are carried out in the CIELAB uniform color space.

2.2 SIMULTANEOUS GMAS

Algorithms belonging to this group map colors so that all of their attributes are changed simultaneously. The algorithms evaluated here do this by moving colors towards a particular point in color space (a center of gravity). For SLIN this center is the point on the lightness axis for which L^*=50 and for the CUSP algorithm the center of gravity is the point on the lightness axis which has the same lightness as the gamut's cusp (the cusp at a given hue angle is the color with the highest chroma). The SLINLLAB algorithm is the same as the SLIN algorithm with the only difference being that it uses the LLAB color appearance model instead of CIE LAB (see Fig. 1). These three algorithms do not have initial lightness compression.

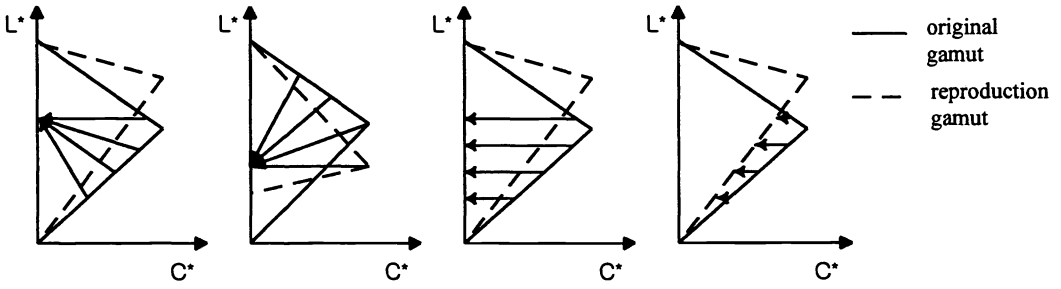


Fig. 1. Overview of successive and simultaneous GMAs.

2.3 PARAMETRIC GMAS

These algorithms change their behavior depending on either the shape of the original and reproduction gamuts at a particular hue angle (as is the case with the Johnson's algorithm) or some other user defined parameters. The Johnson's algorithm evaluated here is adapted from a paper by Johnson et al. [9]. The original algorithm is based on the results of an experiment where the ANSI IT8.7/1 target [11] was given to be reproduced to a number of color reproduction companies in the UK and the USA. The results have shown what gamut mapping was used by scanner operators and the following algorithm is a modification of what has been suggested by the authors.

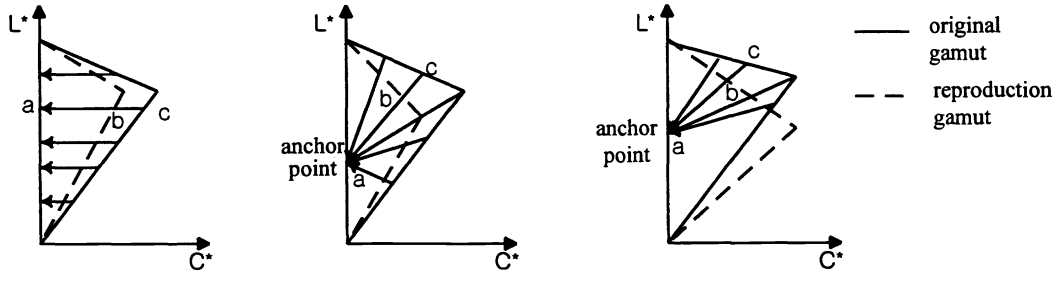


Fig. 2. Overview of Johnson's GMA.

(1) Map the white and black points of the two media onto each other and scale the lightnesses between them linearly (as for the successive GMAs).
(2) Perform additional compression of L* and C* depending on the gamut boundaries of the two media as follows:
   (a) Define the cusp at each hue angle.
   (b) If the L* values of the cusps are similar for both gamuts, map C* along lines of constant L* (see Fig. 2 (a)); else if they are different and one gamut encloses the other, map along the lines going towards the point on the L* axis which is created by the line going through the two cusps (see Fig. 2 (b)); else if one gamut is not enclosed by the other, map towards L*=50 on the L* axis (see Fig. 2 (c)).
   (c) Compression along a given line is determined by x(b-a)/(c-a), where x is the distance along the line from the L* axis, *c* the point on the outer gamut, *b* the point on the inner gamut and *a* the point on the L* axis (see Fig. 2).
(3) Determine the hue shift of the six primaries and secondary colors (dependent on the particular devices used) between the two gamuts and translate the original hues halfway towards the reproduction's hues.

The compression ratios for each of the eight algorithms were always determined along each individual line of compression as shown in Fig. 1 and 2.

3. INTERPOLATION TECHNIQUES

All 3-D interpolation algorithms [12] have the following in common:
- They attempt to approximate a transfer function from the input space to the output space,
- they store in table(s) values for the output function evaluated at discrete points in the input space,
- when presented with a sample point in the input space at which an approximation of the transfer function is desired, they return a value based on a calculation that takes the input point and only a few of the store values.

3.1 TRILINEAR INTERPOLATION

Trilinear interpolation is widely advocated for converting device-independent color representations into device colorant specifications. The form of trilinear interpolation described here produces a continuous output from a continuous input (C^0 continuity in the language of computational geometry), which is an advantage where the table are populated so coarsely that errors exceeding the just-noticeable difference can occur. In trilinear interpolation, the sample function values are arranged into a 3-D table indexed by the independent variables, e.g. x, y, and z. The range of each input variable is usually evenly sampled.

Fig. 3 shows a possible geometric interpretation of trilinear interpolation. The eight stored points form a cube. The point to be interpolated supplies an additional vertex that is used to form eight rectangular hexahedrons, each of which has one of the original vertices and the point to be interpolated as two vertices with sides coincident or parallel to the sides of the original cube. Let us number the eight original vertices sequentially, and let V_i be the volume of the rectangular hexahedron that contains the original vertex directly opposite point i. If the value of the function evaluated at the ith vertex is $\mathbf{p}_i$, the interpolated result is the sum of the volume of each hexahedron, weighted by the value of the function at the corresponding vertex, normalized to the total volume.

$$v = \left(\frac{1}{V}\right)\sum_{i=1}^{8} p_i \times V_i \,, \tag{1}$$

where $V = \sum_{i=1}^{8} V_i$.

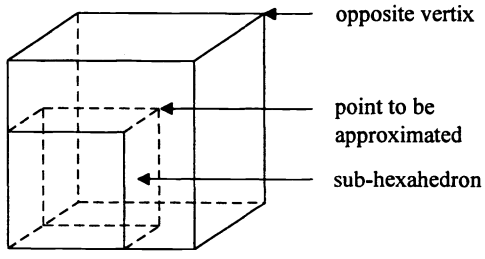


Fig. 3. Trilinear interpolation geometry.

3.2 PRISM INTERPOLATION

Kanamori and others have recently advocated a 3-D interpolation algorithm using six points that they call PRISM interpolation. It may be viewed as a six-point prismatic extraction from the cubical packing. Kanamori claims the PRISM scheme has accuracy approaching that of trilinear interpolation with roughly three-quarters the computational complexity. The extraction of two PRISMs, labeled type 0 and type 1, form a cubical packing, as shown in Fig. 4. The PRISM extraction has a characteristic orientation that can be associated with that of the plane subdividing the two halves of the cube. There are six possible orientations for this plane.

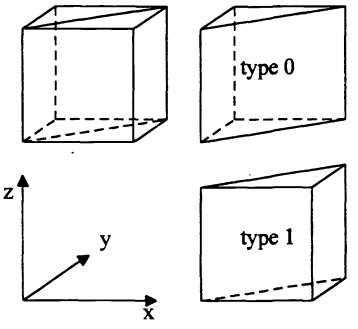
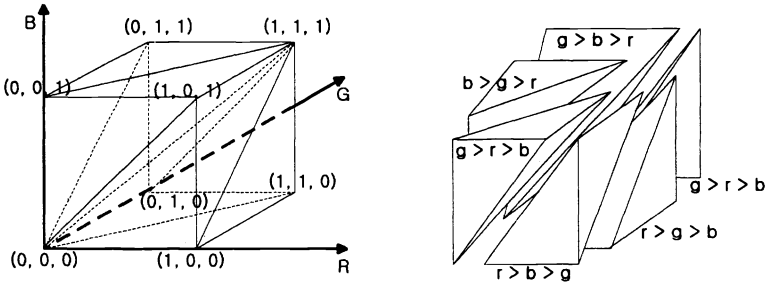


Fig. 4. PRISM extraction.

3.3 TETRAHEDRAL DIVISION AND INTERPOLATION

Tetrahedral division [12] is easier than cubic division because, for example, it is easy to judge inclusion of a set of color values because four surfaces of a tetrahedron are all flat. Therefore, we adapt the tetrahedral division for interpolation. There are many ways to divide a space into tetrahedrons. Fig. 5 shows the tetrahedral division of a subcube. A subcube is divided into six tetrahedrons touching each other at the central axis.



(a) (b)

Fig. 5. Tetrahedral division of a subcube: (a) coordinates indicate the location of points a through h and (b) equations indicate conditions.

3.3.1 CORRESPONDENCE BETWEEN TWO TETRAHEDRONS

A subcube once divided into tetrahedrons in one color space can be linearly related to a point in the corresponding tetrahedron in the other color space. Consider, for example, when one tetrahedron is in RGB space and the other is in the CIEXYZ space. When we pick up a set of tetrahedrons, the relationship of the point inside these tetrahedrons is shown in Fig. 6.

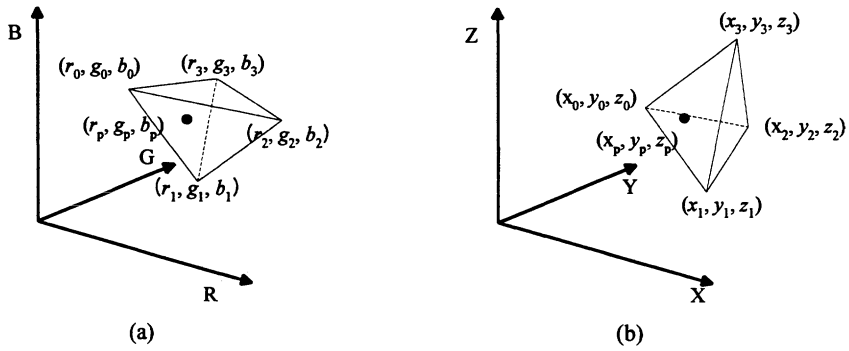


Fig. 6. Corresponding tetrahedrons in two spaces.

3.3.2 CROSSING POINT BETWEEN A TRIABGLE AND A STRAIGHT LINE

A given point is not necessarily included in one of the tetrahedrons that forms a color gamut. For instance, in the calculation from tristimulus values to device system values in a color reproduction device, a desired point to be reproduced may happen to be outside the color gamut. Color reproduction from a CRT to a reflective medium is a typical example, because the color gamut of a CRT is much wider than a reflective medium such as photographic paper. Here, the color values must be placed within the color gamut.

Often a chroma compression technique without changing hue is used. To carry out the calculation, we need to obtain the coordinates of the crossing point between a straight line and the boundary of the color gamut. Fig. 7 shows the geometrical notation of this relationship. The point (x_a, y_a, z_a) is on the gray axis, and the point (x_4, y_4, z_4) is the original point. The line between the two points will be an approximately constant hue locus. The triangle is supposed to be a part of the boundary of the color gamut.

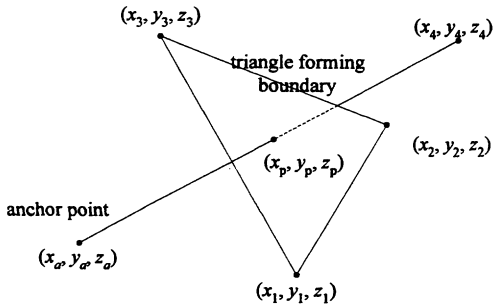


Fig. 7. The crossing of a triangle-forming boundary and a line connecting the target point and achromatic point.

4. THE PROPOSED GAMUT MAPPING USING VARIABLE ANCHOR POINTS OF CONSTANT SLOPE (GMVAPCS)

In the conventional gamut mapping methods, the mapping method toward an anchor point is used. To reproduce maximum chroma on the printer as maximum chroma on the monitor, these methods map along the lines going towards the point on the L* axis which is created by the line going through the two cusps as shown in Fig. 2 (b). But if the L^* values of the cusps are different for both gamuts, the anchor point is far from the center of the L^* axis. In this case, colors mapped in bright region and in dark region show different color change. If the L^* value difference of the cusps is larger, non-uniformity of the color change is increased. In the case of this algorithm using clipping, sudden color change on the gamut boundary of the printer is seriously caused. To solve these problems, anchor point is set at the center of the L^* axis. In this mapping, but, color change in the middle region and in the other region is also different and sudden color change on the gamut boundary is not reduced. And these methods don't consider chroma characteristic to maintain maximum chroma for both gamuts. From these problems, modified adaptive applications of these methods depending on the gamut boundaries of the two medias are used. But problems of both non-uniformity of color change and non-consideration of chroma characteristic are also remained.

To solve these problems, gamut mapping using variable multiple anchor points of constant slope (GMVAPCS) is proposed. Non-uniformity of color change is reduced by using variable multiple anchor points and chroma characteristic is considered by anchor points of constant slope considering direction of the line going through the two cusps. By using this method, sudden color change on the gamut boundary is also reduced. The proposed GMVAPCS performed by lightness mapping and separate mapping methods. In this method, gamut mapping is performed in CIEL*a*b* color space to separate easily lightness and chroma [10]. The gamuts are obtained through the measurements of color samples which have colors covering the majority of color space of each device. From the measurement, the color space conversion is performed using tetrahedral interpolation. In this paper, interpolation from RGB color space to CIEL*a*b* color space is defined as forward interpolation and from the Fig. 6 the equation is expressed as

$$\begin{bmatrix} L^*_p \\ a^*_p \\ b^*_p \end{bmatrix} = \begin{bmatrix} L^*_1 - L^*_0 & L^*_2 - L^*_0 & L^*_3 - L^*_0 \\ a^*_1 - a^*_0 & a^*_2 - a^*_0 & a^*_3 - a^*_0 \\ b^*_1 - b^*_0 & b^*_2 - b^*_0 & b^*_3 - b^*_0 \end{bmatrix} \bullet \begin{bmatrix} r_1 - r_0 & r_2 - r_0 & r_3 - r_0 \\ g_1 - g_0 & g_2 - g_0 & g_3 - g_0 \\ b_1 - b_0 & b_2 - b_0 & b_3 - b_0 \end{bmatrix}^{-1} \bullet \begin{bmatrix} r_p - r_0 \\ g_p - g_0 \\ b_p - b_0 \end{bmatrix} + \begin{bmatrix} L^*_0 \\ a^*_0 \\ b^*_0 \end{bmatrix} \tag{2}$$

The lightness ranges between two gamuts are different as shown in Fig. 8 (a). Therefore, lightness mapping is performed to include the lightness range of input image into printer gamut. To adjust the lightness between two gamuts, the lightness mapping maps lightness linearly so that the minima and maxima of two gamuts are mapped onto each other. The process is expressed as

$$\begin{aligned} L^*_{lp} &= \frac{(L^*_p - L^*_{omin}) \times (L^*_{rmax} - L^*_{rmin})}{(L^*_{omax} - L^*_{omin})} + L^*_{rmin} \\ a^*_{lp} &= a^*_p \\ b^*_{lp} &= b^*_p, \end{aligned} \tag{3}$$

where L^*_{lp}, a^*_{lp} and b^*_{lp} are the result of lightness mapping, L^*_{omax} is maximum lightness of the monitor gamut, L^*_{omin} is minimum lightness of the monitor gamut, L^*_{rmax} is maximum lightness of the printer gamut, and L^*_{rmin} is minimum lightness of the printer gamut.

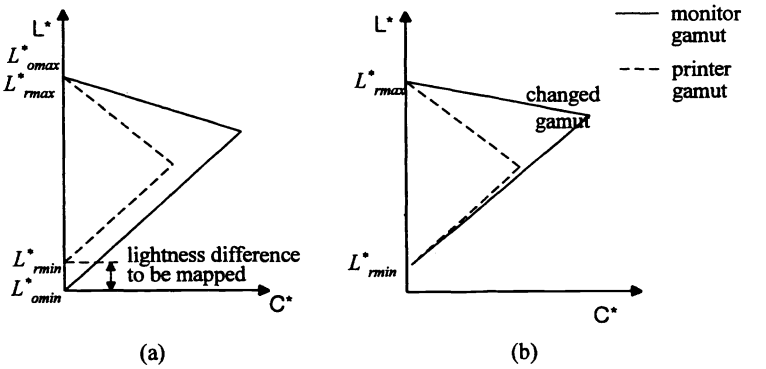


Fig. 8. The lightness mapping.

For the separate mapping, the classification of the regions is set according to the lightness of the monitor and printer gamut. In this paper as shown in Fig. 9, the regions are divided according to the cusp of highest chroma of the two media. Based on these regions, the colors of an input image in bright and dark region are mapped into the printer gamut by clipping their chroma component along the lines of constant slope as shown in Fig. 9. In the case of middle region, mapping toward an anchor point is accomplished.

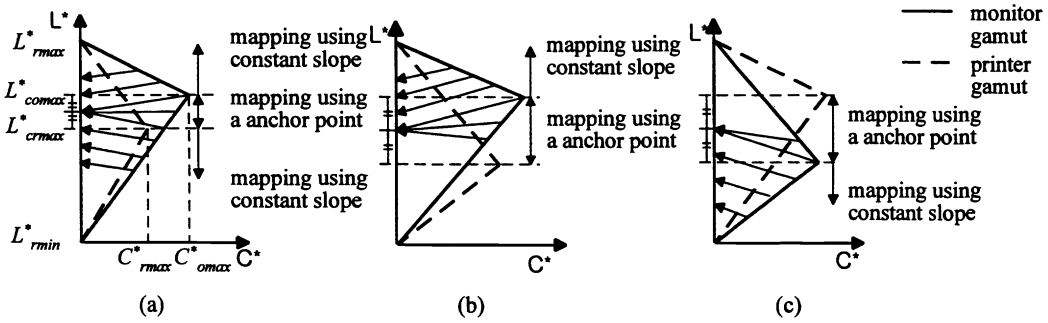


Fig. 9. Gamut mapping using multiple anchor points of constant slope.

In case that the printer gamut is included in the monitor gamut, we can express the anchor point as

$$L^*_a = \begin{cases} L^*_{lp} - \dfrac{(L^*_{comax} - L^*_{crmax})/2}{C^*_{omax}} \times C^*_{lp}, & \text{if } L^*_{comax} \geq L^*_{crmax} \text{ and } L^*_{lp} \geq L^*_{comax} \\ L^*_{lp} + \dfrac{(L^*_{comax} - L^*_{crmax})/2}{C^*_{omax}} \times C^*_{lp}, & \text{if } L^*_{comax} \geq L^*_{crmax} \text{ and } L^*_{lp} < L^*_{crmax} \\ L^*_{lp} - \dfrac{(L^*_{crmax} - L^*_{comax})/2}{C^*_{omax}} \times C^*_{lp}, & \text{if } L^*_{comax} < L^*_{crmax} \text{ and } L^*_{lp} \geq L^*_{crmax} \\ L^*_{lp} + \dfrac{(L^*_{crmax} - L^*_{comax})/2}{C^*_{omax}} \times C^*_{lp}, & \text{if } L^*_{comax} < L^*_{crmax} \text{ and } L^*_{lp} < L^*_{comax} \end{cases}$$

$$a^*_a = 0 \qquad (4)$$

$$b^*_a = 0$$

where L^*_{comax} is the lightness of maximum chroma of the monitor gamut, L^*_{crmax} is the lightness of maximum chroma of the printer gamut, L^*_{lp} and C^*_{lp} are the result of lightness mapping of the input image, and C^*_{omax} is the maximum chroma of the monitor gamut. In the case of middle region, mapping toward an anchor point is proposed. The anchor point is set according to the center lightness of two maximum chroma points as

$$L^*_a = L^*_{comax} - \frac{(L^*_{comax} - L^*_{crmax})/2}{C^*_{omax}} \times C^*_{omax}$$

$$a^*_a = 0 \qquad (5)$$

$$b^*_a = 0$$

Then, the colors outside the printer gamut are clipped toward an anchor point. In case that the printer gamut is partially included in the monitor gamut as in Fig. 9 (b) and (c), the gamut mapping is only accomplished in the included region. The anchor point is expressed as

$$L^*_a = \begin{cases} L^*_{lp} - \dfrac{(L^*_{comax} - L^*_{crmax})/2}{C^*_{omax}} \times C^*_{lp}, & if\ L^*_{comax} \geq L^*_{crmax} \text{ and } L^*_{lp} \geq L^*_{comax} \\ L^*_{lp} + \dfrac{(L^*_{crmax} - L^*_{comax})/2}{C^*_{omax}} \times C^*_{lp}, & if\ L^*_{comax} < L^*_{crmax} \text{ and } L^*_{lp} < L^*_{comax} \end{cases}$$

$$a^*_a = 0 \qquad (6)$$

$$b^*_a = 0$$

From the Fig. 7, the crossing point (L^*_{lp}, a^*_{lp}, b^*_{lp}) can be calculated by the following equations:

$$[a\ b\ c] = [1\ 1\ 1] \bullet \begin{bmatrix} L^*_1 & L^*_2 & L^*_3 \\ a^*_1 & a^*_2 & a^*_3 \\ b^*_1 & b^*_2 & b^*_3 \end{bmatrix}^{-1}, \qquad (7)$$

$$\delta = [a\ b\ c] \bullet \begin{bmatrix} L^*_a - L^*_4 \\ a^*_a - a^*_4 \\ b^*_a - b^*_4 \end{bmatrix}^{-1}, \qquad (8)$$

$$\eta = \frac{1 - [a\ b\ c] \bullet \begin{bmatrix} L^*_4 \\ a^*_4 \\ b^*_4 \end{bmatrix}}{\delta}, \qquad (9)$$

$$\begin{bmatrix} L^*_{lp} \\ a^*_{lp} \\ b^*_{lp} \end{bmatrix} = \eta \bullet \begin{bmatrix} L^*_a - L^*_4 \\ a^*_a - a^*_4 \\ b^*_a - b^*_4 \end{bmatrix} + \begin{bmatrix} L^*_4 \\ a^*_4 \\ b^*_4 \end{bmatrix}. \qquad (10)$$

To print gamut-mapped image, color space conversion is necessary. In this paper, the interpolation from CIEL*a*b* color space to CMY is defined as backward interpolation, and from the Fig. 6 the equation is expressed as

$$\begin{bmatrix} c_{lp} \\ m_{lp} \\ y_{lp} \end{bmatrix} = \begin{bmatrix} c_1 - c_0 & c_2 - c_0 & c_3 - c_0 \\ m_1 - m_0 & m_2 - m_0 & m_3 - m_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \end{bmatrix} \bullet \begin{bmatrix} L^*_1 - L^*_0 & L^*_2 - L^*_0 & L^*_3 - L^*_0 \\ a^*_1 - a^*_0 & a^*_2 - a^*_0 & a^*_3 - a^*_0 \\ b^*_1 - b^*_0 & b^*_2 - b^*_0 & b^*_3 - b^*_0 \end{bmatrix}^{-1} \bullet \begin{bmatrix} L^*_{lp} - L^*_0 \\ a^*_{lp} - a^*_0 \\ b^*_{lp} - b^*_0 \end{bmatrix} + \begin{bmatrix} c_0 \\ m_0 \\ y_0 \end{bmatrix}. \qquad (11)$$

Fig. 10 represents the total procedure of the gamut mapping method. The color space conversion is performed using tetrahedral interpolation. Which can perform with fewer multiplications, an easier coefficient calculation for weighted-averages and a better accuracy in backward interpolation [12,13].

The color image of the RGB format is converted into the CIEL*a*b* color space by forward tetrahedral interpolation. In CIEL*a*b* color space, lightness mapping, anchor point setting, and gamut mapping are performed in sequence. Then the gamut-mapped image is reconverted into CMY color space using backward tetrahedral interpolation.

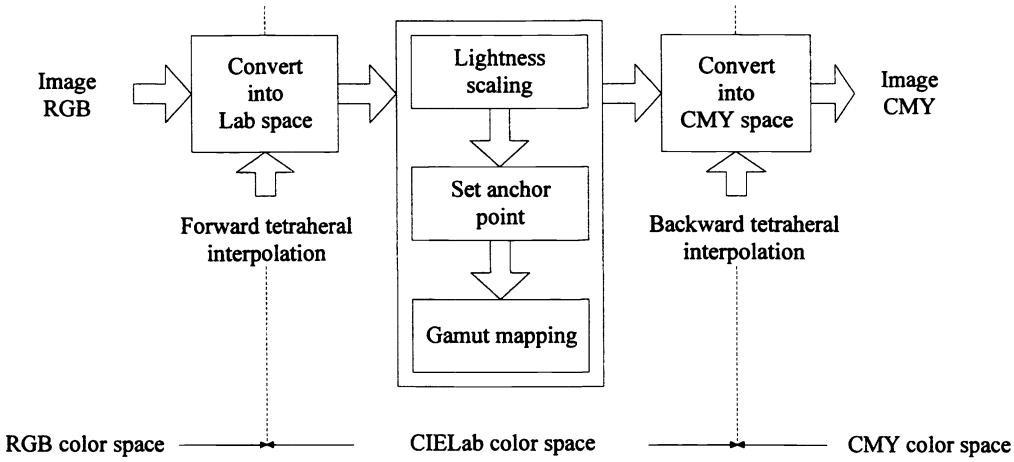


Fig. 10. The total procedure of the proposed GMVAP.

5. EXPERIMENTAL RESULT

Two images were chosen for testing the proposed algorithm. A fresh image was obtained from internet-site and a color chart image was generated from a computer graphic. The original for all images was taken to be their appearance on the calibrated Samsung SyncMaster-700p monitor used throughout the experiment. To print the gamut-mapped images, all images were reproduced on a LG Art-jet printer.

In other to know the gamut of each device, 6×6×6 color samples were generated in RGB space for the monitor and printed in CMY space for the printer. Then, the color samples were measured by spectrophotometer in CIEL*a*b* color space. To measure sample colors displayed on the monitor and printed on the printer, Minolta CA-100 and Minolta CM-3600d were used, respectively. By the device gamuts obtained from the color samples, the gamut mapping was performed.

To compare the GMAs' quality, color difference ΔE^*_{ab} was used. In this process, Macbeth color-chart was used as reference colors. To obtain color difference between the two devices, the Macbeth color-chart was displayed on the monitor and printed on the printer, then the colors were measured by spectrophotometer. From the result, ΔE^*_{ab} was calculated as follows;

$$\Delta E^*_{ab} = \sqrt{(L^*_O - L^*_R)^2 + (a^*_O - a^*_R)^2 + (b^*_O - b^*_R)^2} \tag{12}$$

where $L^*_O a^*_O b^*_O$ is CIEL*a*b* values measured on the monitor, $L^*_R a^*_R b^*_R$ is CIEL*a*b* values measured on the printer. Table 1 shows the comparison of the ΔE^*_{ab}. In the table, the proposed algorithms take less errors than the conventional CUSP's method and Johnson's one.

Table 1. The comparison of ΔE^*_{ab} between colors displayed on the monitor and colors reproduced on the printer.

	CUSP's algorithm	Johnson's algorithm	GMVAPCS
ΔE^*_{ab}	9.86574	13.00708	8.665285

In the Fig. 11 and Fig. 12, the images were printed by error diffusion using various GMAs. Here, (a) is the result printed without GMA, (b) is the result of CUSP's algorithm, (c) is the result of Johnson's algorithm, (d) is the result of the proposed algorithm. In Fig. 11 (b) and (c), the results show that color change in dark region is not discriminated well and colors in bright region are darkening. Fig. 11 (d) using the proposed method shows linear color increment. In Fig. 12, (b) shows cyanic component in the white region and (c) shows blocking effect in the right upper region. But, Fig. 12 (d) don't show cyanic component in the white region and blocking effect. And, the Fig. 12 (d) has higher contrast than (b) and (c), and effectively represent the chroma component. Also, black is more blackish than the conventional methods. Therefore, the image that is reproduced by our proposed algorithms had an almost equivalent visual quality with image displayed on the monitor.

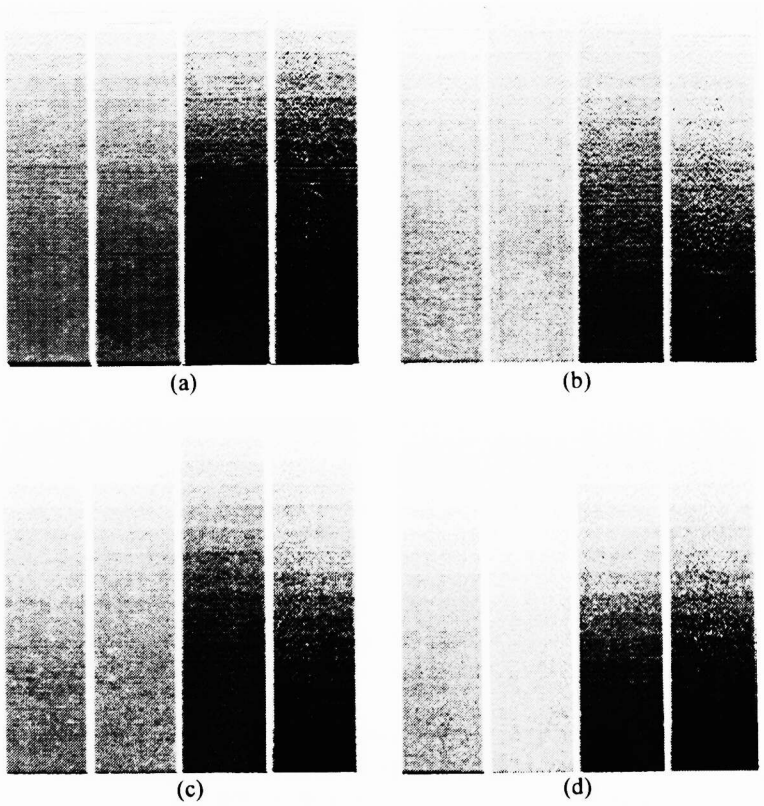


Fig. 11. The graphic images printed by error diffusion using various GMAs. (a) Without GMA (b) CUSP's algorithm. (c) Johnson's algorithm. (d) GMVAPCS.

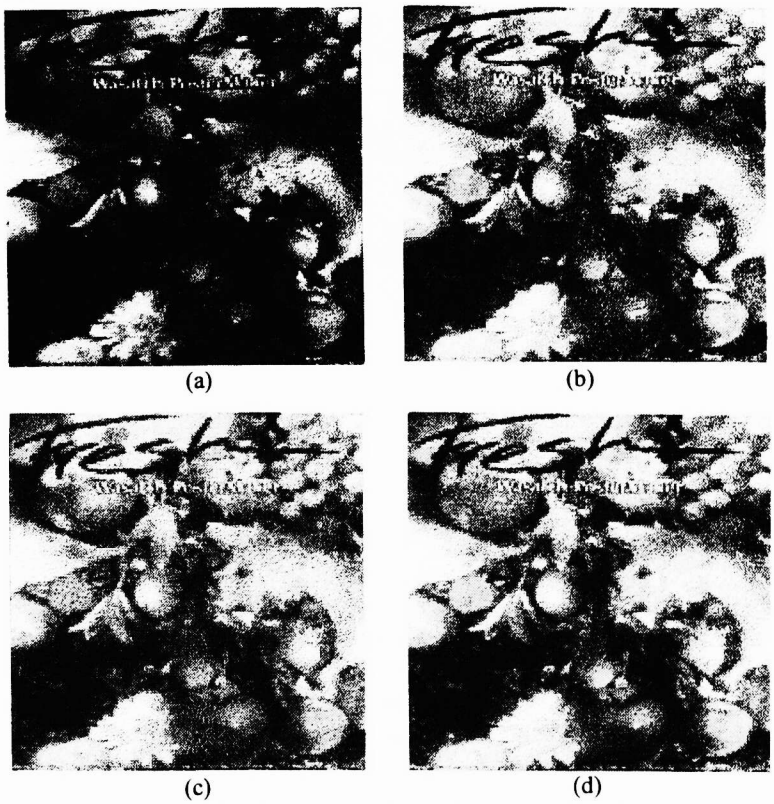


Fig. 12. The real images printed by error diffusion using various GMAs. (a) Without GMA. (b) CUSP's algorithm. (c) Johnson's algorithm. (d) GMVAPCS.

6. CONCLUSION

New method for printing a full resolution color image on limited color output devices was proposed. In this paper, parametric GMA using variable anchor points and clipping was proposed to match images produced on a monitor and a printer. The GMVAPCS used multiple anchor points of constant slope to reduce sudden color change in boundary region between inside gamut and outside gamut. Therefore, the GMVAPCS could reproduce continuous tone image in the boundary region of printer gamut and reduce mapping errors.

REFERENCES

1. M. Ronnier Luo and Jan Morovic, "Two Unsolved Issues in Colour Management - Colour Appearance and Gamut Mapping," *Proceedings of the 5th International Conference on High Technology : Image Science and Technology - Evolution & Promise*, pp. 136-147. Chiba, Japan, 1996
2. Patrick G. Hezog and Michael Muller, "Gamut Mapping Using an Analytic Color Gamut Representation," *SPIE, Device-Independent Color, Color Hard Copy, and Graphic Arts*, February 1997.
3. Patrick G. Herzog and Bernhard Hill, "A New Approach to the Representation of Color Gamut," *The 3rd IS&T/SID Color Imaging Conference, Color Science, Systems and Application*, November, 1995
4. Luo, M. R. and Morovic, J.J. "Two Unsolved Issues in Colour Management - Colour Appearance and Gamut Mapping," *Proceedings of 5th International Conference on High Technology: Imaging Science and Technology - Evolution & Promise*, Chiba, Japan, 1996.
5. Jan Morovic and M.Ronnier Luo, "Cross-Media Psychophysical Evaluation of Gamut Mapping Algorithms," *Proc. AIC Color 97 Kyoto*, 1997
6. Stone, M.C. and Wallace, W.E., "Gamut Mapping Computer Generated Imagery," *Graphics Interface '91*, pp. 32-39, (1991).
7. Laihanen, P., "Colour Reproduction Theory based on the Principles of Colour Science," *IARAIGAI Conference Proceedings Advances in Printing Science & Technology*, vol. 19, pp. 1-36, (1987).
8. Macdonald, L.W., Morovic, J.J. and Saunders, D., "Evaluation of Colour Fidelity for Reproductions of Fine Art Paintings," *Museum Management and Curatorship*, vol. 14, no. 3, pp. 253-281, (1995).
9. Johnson, A.J., Luo, M. R., Lo, M. C., Xin, J. H. and Rhodes, P. A., "Aspects of Colour Management, Part II Characterization of Four-Colour Imaging Devices and Colour Gamut Compression," *Color Research and Application*, vol. 22, pp. 000-000, (1997).
10. Henry R. Kang, Color Technology For Electronic Image Devices, *SPIE Optical Engineering Press*, 1996.
11. MCDOWELL, D., "Summary of IT8/SC4 color activities," *Proceedings of SPIE* 1913, (1993).
12. Po-Chieh Hung, "Colorimetric Calibration in Electronic Imaging Devices Using a Look-up-Table Model and Interpolation," *Journal of Electronic Imaging*, Vol. 2, No. 1, pp. 53-60. January 1993.
13. Gabriel Marcu and Satoshi Abe, "CRT and Ink Jet Printer Models for Device Independent Color Reproduction in Image Transmission," *Proceedings of The 2nd IS&T Color Imaging Conference, Color Science, Systems and Application*, November, 1994.