# CRT and Ink Jet Printer Models for Device Independent Color Reproduction in Image Transmission

*Gabriel Marcu and Satoshi Abe**
*Array Corporation, Tokyo, Japan*
** Department of Information Science, Faculty of Science, University of Tokyo, Japan*

## Abstract

This paper proposes an implementation of tetrahedral interpolation based on a nested structure of look-up tables for fast device dependent/independent color transformation. The CRT and ink jet printer models were used for exemplification and the procedure was used for colorimetric calibration in image transmission, but the method is generally applicable. The models are based on look-up tables and tetrahedral interpolation of LUT values. The direct transformation avoids completely the searching procedure for the tetrahedron that includes the interpolated color. For the reverse transformation, the searching procedure is applied to a reduced set of tetrahedrons and is optimized based on the use of the nested tables structure. The time performance of algorithm increases with the size of LUT, and varies from 8 times for 5×5×5 LUT to 4000 for 33×33×33 LUT. The color gamuts of the two devices are represented and compared in Lab and Luv color spaces in order to select the optimum space to perform the mapping procedure of outside gamut colors.

### Color in Image Transmission

In image transmission, the accuracy of color reproduction is an important problem. A received digital signal of one image can be decoded and displayed or printed as an array of pixels, each pixel specifying a color. Usually the color is specified in a device dependent color space, RGB or YMCK. In case of device dependent specification of color, the reproduced color may differ from device to device, even if the same numeric specification of color is used. The differences in color appearance on different devices are caused by many factors: principle (additive or substractive) of color rendition, technology, materials, parameters of the color rendition mechanism, device type, device variability, ambient observation conditions[1,2].

For colorimetric accuracy in color reproduction, transformations are required between color specification for different devices. The complexity and the diversity of color reproduction devices makes more difficult to adapt the analytical available models[3,4] for calibration process. More attractive in terms of adaptability to any new device as well as for modelling the existent devices appear to be the empirical models based on translation tables (LUT)[5]. The LUT to implement all possible correspondences between two device gamuts results impractical large size. Therefore, a limited size of LUT is preferred and the interpolation techniques for deriving the intermediate values are used. Different interpolation techniques were proposed[6-9] and due to tridimensionality of the color spaces, tetrahedral interpolation techniques are preferred[9]. The calibration process requires to derive the LUT contents performing colorimetric measurements on color samples reproduced on devices in question.

This paper proposes a fast implementation of tetrahedral interpolation technique based on a nested structure of translation tables. The transformation were discussed from device dependent color space to a device independent color space, where the color gamuts can be compared. Both direct and inverse transformation are discussed in order to cover also the case of device to device direct transformations. The technique is applied to CRT and ink jet printer devices, but it is not limited to them. The procedure was simulated using hypothetical measured data, that were derived using the analytical transformations, instead to be resulted from direct color samples measurements. For CRT, the linear transformation[10] from RGB to XYZ was used. For ink jet printer, the Neugebauer model[3], with additional corrections[4], was used for transformation from YMCK to XYZ. Additionally, the transformation from XYZ to Lab or Luv[2] independent color spaces was used. The section 2 and 3 presents the implementation of the interpolation technique. The representation of CRT and ink jet printer color gamuts is presented in section 4 and an analysis concerning the optimal space for performing the mapping procedure of outside gamut colors is provided.

### The Implementation of Interpolation Technique for Direct Transformation

The paralelipiped volume of the device dependent color space is partitioned by dividing each component axis into N intervals that correspond to N+1 points. The corresponding LUT, named ColorLUT, result in $(N+1)\times(N+1)\times(N+1)$ cells that store the colors values, $\mathbf{C}_i$, corresponding to the points, $\mathbf{P}_i$, of input values. The dividing structure is regulate and corresponds to cubic decomposition of the device space.

The interpolation is based on the diagonal tetrahedral decomposition[6,9,10,11] of the elementary cubes that results during generation of the LUT color table. We select the diagonal tetrahedral decomposition, but the algorithm that we propose works also for other tetrahedral partition of the device space. Figure 1 illustrates the diagonal partition process.

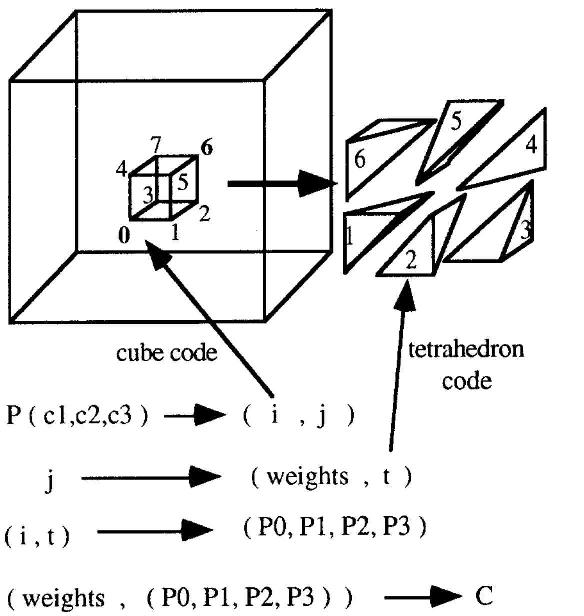



*Figure 1. The diagonal tetrahedron partition and the illustration of nested tables procedure for fast interpolation*

The tetrahedral interpolation procedure, is based on the following steps in order to find the representation, **C**, in the device independent color space, of the device dependent specified color, **P**:

1. find the tetrahedron, $\mathbf{P_0,P_1,P_2,P_3}$ that encloses the point **P**;

2. find the interpolation weights[3,4], $0\leq a_1, a_2, a_3 \leq 1$ such that:

$$\mathbf{P} = \mathbf{P_0} + (\mathbf{P}_1 - \mathbf{P_0}) . a_1 + (\mathbf{P}_2 - \mathbf{P_0}) . a_2 + (\mathbf{P_3} - \mathbf{P_0}) . a_3 \quad (2)$$

3. interpolate the unknown color **C** using the color values corresponding to the tetrahedron vertices, $\mathbf{C_0,C_1,C_2,C_3}$:

$$\mathbf{C} = \mathbf{C_0} + (\mathbf{C}_1 - \mathbf{C_0}) . a_1 + (\mathbf{C}_2 - \mathbf{C_0}) . a_2 + (\mathbf{C_3} - \mathbf{C_0}) . a_3 \quad (3)$$

The conventional interpolation algorithm requires to perform the color computations for solving the system (2) for each color pixel of the image. The first and second step can be performed simultaneously, since the inclusion condition of color **P** within the tetrahedron, $\mathbf{P_0,P_1,P_2,P_3}$ can be verified by searching the values $a_i$, such that $0 \leq a_i \leq 1$. It can be observed that in this form, the algorithm is time consuming, requiring to solve at least one time the matrix equation (2), and for this reason some authors[5,7] recommend a presearching procedure for decreasing the computation time.

In order to avoid the time consuming searching routine and all corresponding computations, we propose to use a method based on addressing of precomputed allocation tables. The idea of the method is based on the observation that the weights, $a_1, a_2, a_3$, depends only on the relative position of point **P** with respect to the vertices $\mathbf{P_0,P_1,P_2,P_3}$, which form a regulate and repetitive structure, and not on the absolute coordinates of $\mathbf{P_0,P_1,P_2,P_3}$. More clearly, if the tetrahedrons in a cube are coded from 1 to 6 as in figure 1, for all points **P** that have the same relative position with respect to the vertices of the tetrahedrons with same code, it corresponds a unique weights set, independently on the tetrahedron position in the space.

Two indexes, i, j, were derived from the color components $c_1, c_2, c_3$ of the color **P**. The first index selects the vertex 0 of the elementary cube that contains the point **P**, and the second index, j, selects the tetrahedron code, t, that contains the input color **P**. The tetrahedron code, t, is stored in the first of the nested tables, named teTB, as it appears in figure 2.

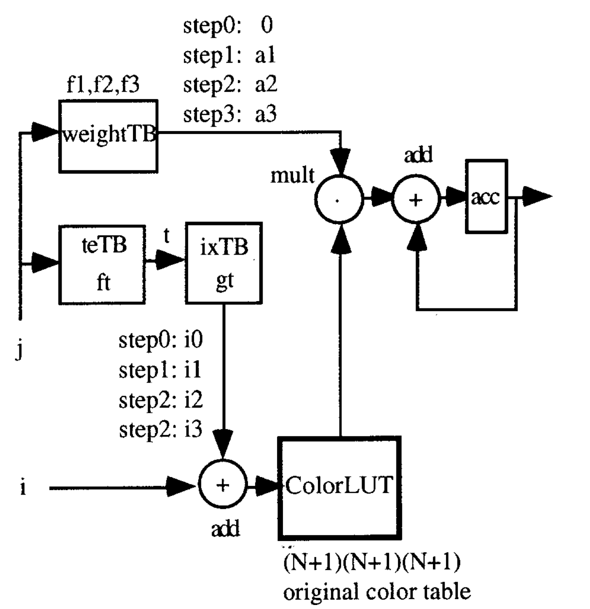



*Figure 2. The diagram of the nested tables implementation of the tetrahedral interpolation*

It can be observed that the second index can specify the relative position of **P** inside the tetrahedron, and thus it can simply point into a table, named weightTB, that contains the weights, $a_1, a_2, a_3$, derived from equation (2). The tetrahedron code, t, can select in a second nested table, named ixTB, the relative positions of the vertices $\mathbf{P_0,P_1,P_2,P_3}$, with respect to the vertex 0 of the cube (fig.1), pointed by the first index, i. The relative positions of the vertices $\mathbf{P_0,P_1,P_2,P_3}$ are denoted by the offsets i0,i1,i2,i3, with respect to the first index, i. Note that for diagonal decomposition, two of four vertices of the tetrahedron are in all cases in the vertices 0 and 6 of cube. What it remains is to address the ColorLUT table with the global index i+i1, i+i2, i+i3 to find the values required for interpolation and to perform the computations required by equation (3). The indexes i0,i1,i2,i3 can be selected according to the preferred tetrahedral decomposition of the cube.

In the following, the equations of the proposed algorithm are summarized. For simplicity of presentation, we assume that N, the number of samples per each component of input value, divides the maximum value that codes these components, L. Usually, L and N are power of 2. The algorithm runs in 4 steps, each one corresponding to each terms of the equation (3).

Notation:

$\mathbf{P} = [\ c1,c2,c3\ ]^T$ , where ci = 0,1,...,L ; A = In( L/N ) = L/N , where In(a) represent grater integer smaller than a.

ColorLUT table size : [ (N+1)(N+1)(N+1) ];
weightTB size : [ A.A.A ][3] ;
ixTB size : [6][4] ;

first index, i :

i = N.N.In( c3/A ) + N.In( c2/A ) + In( c1/A ) ;

second index, j :

j = A. A .( c3 - A.In(c3.L/N) ) +
+ A .( c2 - A.In(c2.L/N) ) +
+ ( c1 - A.In(c1.L/N) ) ;

t = teTB( j );
**C** = 0 ;

step k ( k = 0,1,2,3) :

$i_k$ = ixTB[ t ][ k ] ;
$\mathbf{C_i}$ = ColorLUT ( i + $i_k$ ) ;
$a_k$ = weightTB [ j ][ k ] , $a_0 = 1$ ;
**C** = **C** + $a_k$ . $\mathbf{C_i}$ ;

The algorithm is performed only once for each pixel and the time consuming searching procedure is eliminated, as well as all computation required to solve the equation (2). All computations required to fill the tables weightTB, teTB, ixTB are performed a single time, at initialization, and for interpolation process the tables are only addressed, following the diagram in figure 2. The proposed nested tables procedure enables hardware implementation.

### The Implementation of Interpolation Technique for Reverse Transformation

The inverse transformation requires the same steps as direct transformation, except that the $\mathbf{C_i}$ color values and color **C** are used to determines the interpolation weights. The steps required in reverse transformation are:

1. find the tetrahedron, $\mathbf{C_0,C_1,C_2,C_3}$ that encloses the point **C**;

2. find the interpolation weights, $0 \le a_1$ , $a_2$ , $a_3 \le 1$ such that:

$$\mathbf{C} = \mathbf{C_0} + (\mathbf{C}_1 - \mathbf{C_0}).\ a_1 + (\mathbf{C}_2 - \mathbf{C_0})\ a_2 + (\mathbf{C_3} - \mathbf{C_0}).\ a_3 \quad (4)$$

3. interpolate the unknown color **P** using the color values corresponding to the tetrahedron vertices, $\mathbf{P_0,P_1,P_2,P_3}$:

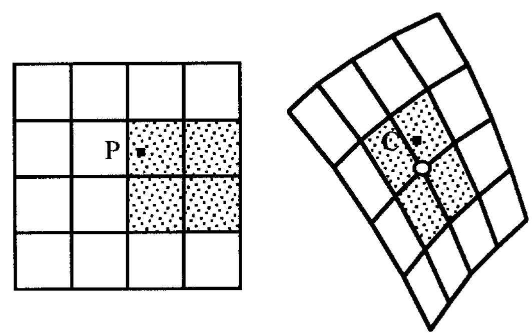
$$\mathbf{P} = \mathbf{P_0} + (\mathbf{P}_1 - \mathbf{P_0})\ .\ a_1 + (\mathbf{P}_2 - \mathbf{P_0})\ a_2 + (\mathbf{P_3} - \mathbf{P_0}).\ a_3 \quad (5)$$

The problem is more complex since the $\mathbf{P_0,P_1,P_2,P_3}$ uniform structure N × N × N in device dependent color space is projected in a $\mathbf{C_0,C_1,C_2,C_3}$ not uniform lattice, in device independent color space.

In order to find the tetrahedron of the color space that contains the color to be reverted from independent device coordinates to device dependent coordinates, a time consuming searching procedure running for all the set points of the lattice is recommended[3,5].

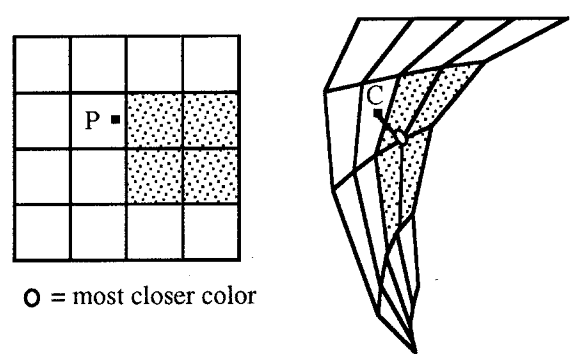
In our model, we try to optimize the searching procedure, eliminating as much as possible from the tetrahedron candidates. We proceed in the following steps:

1. determine a reduced set **S** of tetrahedrons that are candidates for possible inclusion of color **C**. Note that tacking into consideration only the tetrahedrons with a vertices in most closer color $\mathbf{C_k}$ is not safe. An example that illustrates a situation when this procedure fails is offered in figure 3b, that corresponds to the bidimensional case (for clarity of the diagram).




device dependent space device independent space

*a. Example when selection of 4 neighbor cubes of most closer color contain the color to be interpolated*




device dependent space device independent space

*b. Illustration of principle of neighbors of a single color, selected color, selected closer to investigated color C*

In our method we build the set **S** of tetrahedron candidates using the following algorithm:

1.a. determine the maximum length, lmax, of the all tetrahedrons edges;

1.b.determine the first 8 closer colors $\mathbf{C}_{i1...8}$ to the color **C** , and retains only that colors that have the distance to **C** smaller than maximum edge length, lmax, of possible tetrahedrons; the resulted set of points is referred by subscripts i1,i2,...,i8; Note that in this step, if the color **C** is an outside gamut color, the result can be a null set of points;

1.c include in the set **S** all the tetrahedrons that have one (or more) vertices in colors $\mathbf{C}_{i1...8}$ ; For diagonal decomposition, the tetrahedrons are contained in the cubes that contains the colors $\mathbf{C}_{i1...8}$ as vertices. The numbers of tetrahedrons in **S** depends on the position of colors $\mathbf{C}_{i1...8}$ in the device color gamut. In case that the colors $\mathbf{C}_{i1...8}$ are all within the gamut and not on the boundary gamut, the maximum number of tetrahedrons in **S** is M = 48. In case when the colors $\mathbf{C}_{i1...8}$ are on the boundary gamut, the number of tetrahedrons is smaller.

2. search what tetrahedron in set **S** includes the color **C** ; This step appears computational, requiring to solve the equation (4), for 48 times in the worst case. The corresponding 4 vertices of each tetrahedron from the 48 possibilities are derived using the nested tables structure introduced for direct transformation. Each index i1,...,i8 is placed successively instead of first index, i, to address the ColorLUT table. For each index i, all six possible tetrahedron codes, t=1,2,...,6 are generated in order to obtain the six tetrahedrons corresponding to color i. For one index i, and one code t, the four steps are required to determine the tetrahedron vertices, $\mathbf{C_0},\mathbf{C_1},\mathbf{C_2},\mathbf{C_3}$. If the procedure uses the indexes i1,...,i8 in an order corresponding to the increasing distances to the investigated point **C**, the chance to discover the tetrahedron containing the color **C** in the first steps of searching process is increased, and no 48 cases are searched.

It can be observed that the computation time, is reduced in a ration 6.N.N.N / 48 = $(N/2)^3$, with respect to the searching procedure that investigates all tetrahedrons. Table 1 summarizes the improvement factor for computation time of inverse transformation. It can be observed that the improvement performance vary from 8:1 in case of 5×5×5 LUT to 4000:1 for 33×33×33 LUT which represents a significant improvement.

**Table 1.**

| LUT size | improvement ratio |
|---|---|
| 5×5×5 | 8 |
| 9×9×9 | 64 |
| 17×17×17 | 512 |
| 33×33×33 | 4096 |

In case when no tetrahedron satisfy the inclusion condition, or in case when all neighbors are at a higher distance than maximum edge length of all tetrahedrons (step 1.b results in a null set of points), the color **C** is treated as an outside gamut color, and a procedure described in the next paragraph is used.

### Mapping the Outside Gamut Colors

The color gamut of CRT and of the ink jet printer devices differ. In general case, the ink jet printer has a smaller color gamut than the CRT display. Figures 5 and 6 offer an example of color gamut representation in Lab and Luv color spaces for CRT and ink jet printers that we used (SuperMac Dual Mode Display and Canon Pixel Jet printer), based on the models described in this paper. The representations are obtained by the transformation of the vertices of device dependent RGB and YMCK respectively, to Lab and Luv color spaces. The same drawing parameters (scale, view angle, number of segments per edge) of the two color diagrams were used in order to get a comparable representation of the two gamuts.

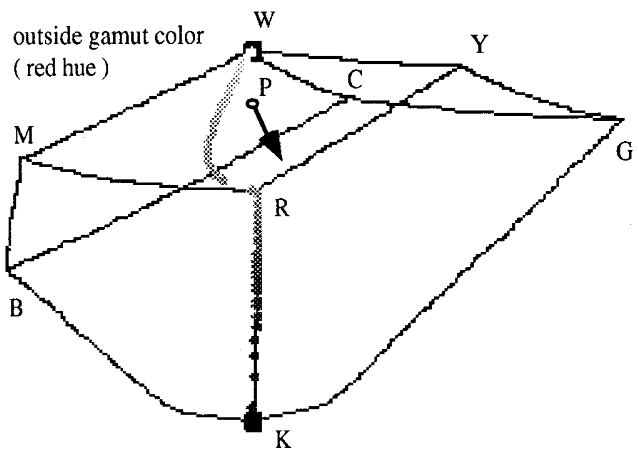
Since the differences of the color gamuts must be accepted, color distortions are introduced when same image data are represented on different color reproduction devices, due to approximation of the outside gamut colors.

Usual procedure of mapping of outside gamut colors uses a constant hue section of the gamut and select the color at the cross point between the gamut boundary and one straight line contained in the constant hue section that connects the investigated color and an achromatic point on the lightness axis of the gamut.

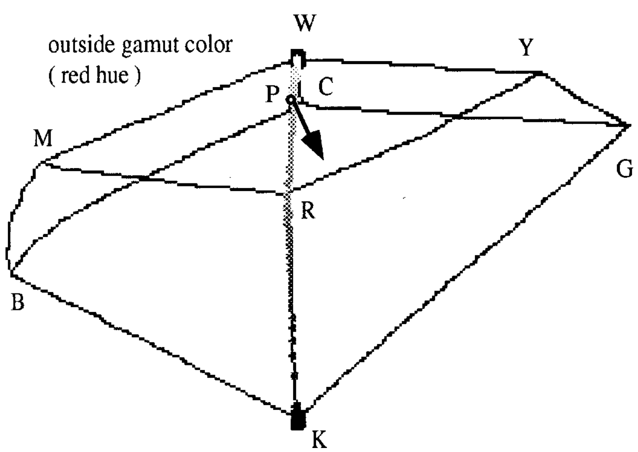
In order to find the best procedure of mapping of the outside gamut colors, we investigate the representation of gamuts in Lab and Luv color spaces. For each gamut we represents the locus of colors with constant hue, maximum saturation and variable lightness. For exemplification, in figures 5 and 6 the locus corresponding to red hue was drawn. The locus is represented as a gray line on the gamut boundary from black to white point.

Figure 5a represents the CRT gamut in Lab color space. It can be observed that the locus is not contained in a planar section containing the luminance axis of the color gamut, and its shape results as a curve surface. This seems to be normal since the Lab color space was originally introduced for substractive devices[12]. Performing the mapping process for outside gamut colors in a plane of constant hue of the Lab space, that is not a constant hue for device gamut will conduct to errors. It can be concluded that for CRT the Lab space is not suitable for constant hue mapping procedure of outside gamut colors. It is also important to note that the minimum distance to the color gamut, corresponding to the minimum Lab color difference does not produce the best mapping procedure. Figure 5a illustrates an example of projection of a point corresponding to an outside gamut red hue color, normal to the surface of the color gamut as it is estimated from the proposed color model, showing the hue error introduced when the projected point is selected to represent the output gamut color.

Figure 5b represents the CRT gamut in Luv color space, and for this representation it can be concluded that the constant hue locus corresponds to the vertical section of the color space. As a conclusion, we select the Luv gamut representation in order to perform the mapping procedure of outside gamut colors for the CRT device.




a. Lab representation of SuperMac Dual Mode CRT gamut

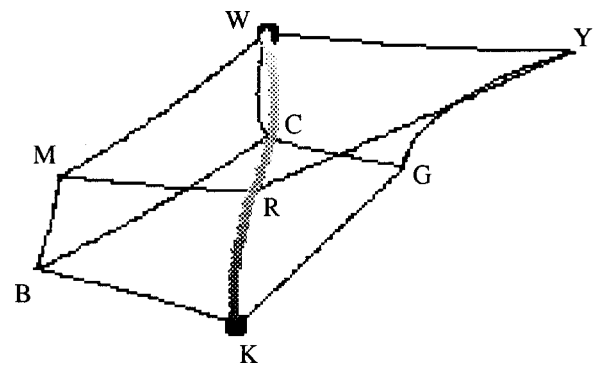



b. Luv representation of SuperMac Dual Mode CRT gamut

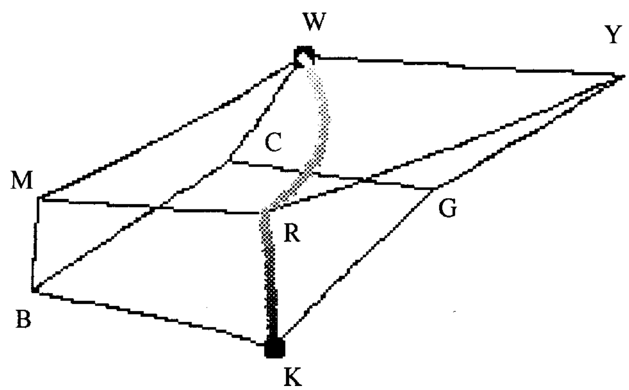
*Figure 5. The Lab and Luv representation of displayable color gamut corresponding to SuperMac Dual Mode CRT display. The gray line represents the locus of the constant hue (red) and maximum saturation gamut colors.*

The same discussion can be carried out for the representation of ink jet printer gamut in the Lab and Luv color space. Figure 6a and 6b show the Lab and Luv gamut corresponding to ink jet printer. It can be observed that the locus of device constant hue is not exactly contained in constant hue vertical section of Lab space, but the plane shape approximation is far better than the corresponding representation in Luv color space.

As a conclusion of CRT and ink jet printer gamut representations, we select the Luv color space for CRT and Lab color space for ink jet printer in order to perform the mapping procedure for outside gamut colors. In the selected spaces, the constant hue is represented by a straight line that connects the outside gamut color with an achromatic point on the luminance axis. The achromatic point can be selected on luminance axis accordingly to different criteria, but we preserve hue and lightness of outside color gamut and select the saturation such that the resulted point to be within the color gamut.




a. Lab representation of Canon Pixel Jet gamut




b. Luv representation of Canon Pixel Jet gamut

*Figure 6. The Lab and Luv representation of printable color gamut corresponding to Cannon Pixel Jet Printer. The gray line represents the locus of the constant hue (red) maximum saturation gamut colors.*

## Conclusions

This paper proposed an implementation of tetrahedral interpolation for fast color transformation used in colorimetric devices calibration. The nested allocation tables structure enables to increase the time performance of the interpolation algorithm from 8 to 4000 times according to capacity of used LUT. The interpolation procedure using the nested allocation tables is generally applicable to all tetrahedral interpolations and is not limited to the diagonal interpolation used in the paper. The Lab and Luv color spaces were compared in order to select the optimum representation for performing the mapping procedure for outside gamut colors. It was found that the Luv for CRT and Lab for ink jet printer device minimizes the hue error when the estimate color is selected on segment connecting the outside gamut color to an achromatic point of the space.

## References


1. R. W. G. Hunt, The Reproduction of Colour, Fountain Press, England, 1987.
2. G. Wyszecki, W. S. Stiles, Color Science: Concepts and Methods, Quantitative data and Formulae, John Wiley & Sons, 1982.
3. H. E. J. Neugebauer, Die theoretichen Grundlagen des Mehrfarbenbuchdrucks Zeitschrift für wissenschaftliche Photographie, Photophysik und Photochemie, V36, N4, p.73, April 1937, reprinted in Neugebauer Memorial Seminar on Color Reproduction, SPIE V1184, p194, December 1989.
4. G. G. Field, Color and Its Reproduction, Graphic Arts Technical Foundation, GATF, 1988.
5. P.C.Hung, Colorimetric calibration in electronic imaging devices using look-up table model and interpolations, Journal of Electronic Imaging, V2, N1, January 1993.
6. K.Kanamori, H.Kotera, Color Correction Technique for Hard Copies by 4 Neighbors Interpolation Method, Journal of Imaging Science and Technology, February 1992, V36, N1, p.73.
7. K. D. Genetten, RGB to YMCK conversion using 3D Barycentric Interpolation, *SPIE Electronic Imaging Conference*, San Jose, 1993, Vol. 1909 - Device Independent Color Imaging, p117.
8. D. A. Clark, D. C. Strong, T. O. White, Method of Color Conversion with Improved Interpolation, U.S. Patent 4,477,833, Oct.1984.
9. J.M.Kasson, W.Plouffe, S.I.Nin, A tetrahedral Interpolation Technique for Color Space Conversion, *SPIE Electronic Imaging Conference*, San Jose, 1993, Vol. 1909-Device Independent Color Imaging, p.127.
10. G. W. Meyer, D. P Greenberg, Perceptual color spaces for computer graphics, *Color and the computer*, edited by H. J. Durrett, Academic Press, Inc., 1987, p.83.
11. L. G. Thorell, W. J. Smith, Using Computer Color Effectively, HP, Prentice Hall, Englewood Cliffs, 1990, p.181.