RC22388 (W0204-001) April 1, 2002
Computer Science

IBM Research Report

Derivation of efficient color-space conversion formulae for n-dimensional interpolation

Gordon W. Braudaway
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Derivation of efficient color-space conversion formulae for *n*-dimensional interpolation

Gordon W. Braudaway
International Business Machines Corporation, Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT

Conversions from one color-space to another are frequently used in image processing. Many conversion methods have been described mathematically, a few of which have been elevated to be international standards. However, for some conversions, purely mathematically based methods have been found wanting, and the method of *measurement and interpolation* has been used to produced more accurate results. Nowhere is this method more prevalently used than in dealing with colors resulting from combinations of inks or toners applied to paper. Although ink or toner colors of Cyan, Magenta, Yellow and Black are most commonly used, the use of a larger number of inks or toners that expand the color gamut is becoming more important for high fidelity color printing. The subject of this paper is the derivation of true *n*-dimensional linear interpolation formulae that are much more efficient than "tri-linear" or "quad-linear" and that can be used to convert from one *n*-dimensional color-space to another of equal, fewer or greater dimensions. The mathematical principle to be used for deriving the formulae is called *axiom based induction*. An interesting application of these formulae might be the conversion of a seven-dimensional color-space to a four dimensional color-space that would allow a seven-color *master image* to be "re-purposed" for printing by a less costly four-color method. Another application might be the use of a seven-dimensional interpolation, applied iteratively to produce a "corner turn," that could allow the direct mapping of three-dimensional color into seven-dimensional ink densities. An example of interpolation errors resulting from round-trip conversion of three to four and back to three dimensions will be given.

Keywords: color-space conversion, *n*-dimensional interpolation, color image processing

1. INTRODUCTION

Conversions from one color-space to another are frequently used in image processing. Many conversion methods have been described mathematically, a few of which have been elevated to be international standards. However, for some conversions, purely mathematically based methods have been found wanting, and the method of *measurement and interpolation* has been used to produced more accurate results. Nowhere is the method of measurement and interpolation more prevalently used than in dealing with printed colors, that is, colors resulting from combinations of inks or toners applied to paper. One conversion of considerable interest supports displaying of printer-ready color images on a Cathode Ray Tube (CRT) or a Liquid Crystal Display (LCD) [1,2]. It is sometimes called "soft proofing." For soft-proofing to be of greatest value, the calibrated LCD- or CRT-displayed image must have sufficient color fidelity that, when viewed side-by-side with a printed copy of the same image under controlled lighting, the two images will be essentially indistinguishable from one another. Equally importantly, for colors that lie in the intersection of the gamuts of the printer and display, the two images must be measurably indistinguishable, varying only in luminance if at all.

A printer-ready image is defined herein as having the color of each of its pixels separated into three, four, or as many as seven primary color components. The primary color components represent the spatial densities of the inks or toners to be applied to paper using halftone screens. Although ink or toner colors of Cyan, Magenta and Yellow (CMY) or Cyan, Magenta, Yellow and Black (CMYK) are most commonly used, the use of a larger number of inks or toners that expand the color gamut is becoming more important for high fidelity color printing. A straightforward process for color conversion might be to print color patches of all possible ink or toner combinations and to measure them. The method of color conversion could then be simply a four-dimensional "table look-up." But this method is entirely impractical. A CMYK color image in TIFF format specifies 256^4 (or 4,294,967,296) combinations of dye or ink densities, each combination producing a color slightly different from its neighbors when applied to paper. The practical solution is to form a regular but coarser grid of ink or toner density combinations, to print and measure the far fewer combinations, and to interpolate among the measured colors to determine the intermediate values. The tessellated subspaces or volumes of the color-space have as their vertices the coordinates of the grid points in the coarse grid.

The subject of this paper is the derivation of efficient *n*-dimensional linear interpolation formulae (also called volumetric interpolation) that can be used to convert printer-ready images having three to seven color components per pixel into a three-dimensional color-space, or more generally for converting from one *n*-dimensional color-space to another of equal, fewer or greater dimensions. The conversion form CMYK to RGB display primaries, for instance, requires a four-dimensional interpolation applied three times to achieve its result. The same four-dimensional interpolation formulae, used iteratively, can execute a "corner turn"; that is, it can be used to produce the inverse of the CMYK to CIELAB L*a*b* calibration table that defines the transformation of a regular grid of L*a*b* values to CMYK ink or toner densities that will produce the same color.

The mathematical principle to be used for deriving the formulae for high-order interpolation is called *axiom based induction*. Axioms, or rules, are self-evident postulates that form the foundation for mathematically rigorous proofs. The axioms that will be stated can be applied in two-dimensional space where they can be verified by inspection, extended by induction to three-dimensional space, where they can also be verified by inspection of a physical cube, and then extended by induction to higher dimensional *hyperspaces* where physical examination is no longer possible and rigorous proofs become more difficult. The verified axioms of induction serve as a compass for navigating in these higher dimensional hyperspaces and for deriving the formulae for interpolation. Using axiom based induction in this manner always contains the risk that the group of axioms, seemingly complete for deriving interpolation formulae in two and three-dimensional spaces, may unknowingly turn out to be incomplete when applied in hyperspaces having more than three dimensions. This risk, however, can be mitigated, at least in a pragmatic sense, by verifying the interpolation formulae for a very large, but finite, set of points while awaiting formal proofs of complete filling[3,4] of the hyperspace from the mathematicians.

2. FUNDAMENTALS OF INTERPOLATION: THE TWO-DIMENSIONAL SPACE

In all discussion that follows, the *n*-dimensional "volume" will always have unit dimensions; that is, its edges will lie in the domain (0,1), and all coordinates of the vertices of the volume will be zero or one. This is not a restriction in generality. It is a straightforward process to locate a particular tessellated volume in the color-space that surrounds the *point of interest* using only the coordinates of the regular grid points and the coordinates of the point of interest. The point of interest is herein defined to be the point within the volume at which an interpolated dependent value is to be evaluated, and the coordinates of the point of interest are the independent values of that evaluation. The coordinates of the vertices of the tessellated volume surrounding the point of interest, and the coordinates of the point of interest, can easily be linearly mapped to the domain (0,1) before interpolation is begun.

The establishment of axioms will be started using a two-dimensional volume that is a square with unit sides. Of course, in two dimensions the "volume" is actually a plane and its associated "vertices" are simple corners. It can be approximated by a square drawn on a sheet of paper. This leads to the first axiom.

> **Axiom #1: The number of vertices in an *n*-dimensional unit volume is 2^n, and the vertices are numbered using their binary coordinate values.**

For a unit square, as shown in Figure 1, the two coordinates of its vertices, or corners, are *x* and *y.* According to Axiom #1, the vertices, numbered in *xy* coordinate order, are 00_2, 01_2, 11_2 and 10_2. The subscript, 2, designates a base-2 or binary number having digit values of {0, 1} only. Correspondingly, the number C_{16} designates a base-16 or hexadecimal number having digit values {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}. A hexadecimal digit is precisely equal to four binary digits, so $C_{16} = 1100_2$. This level of detail may seem somewhat labored at this point but it will pay off handsomely when we move blindly into hyperspaces having more than three dimensions.

Every vertex in the volume can be connected to every other vertex, and these connections are either edges or diagonals. For example, vertex 00_2 can be connected to the other vertices by diagonals or edges $10_2 : 00_2$, $11_2 : 00_2$, and $01_2 : 00_2$.

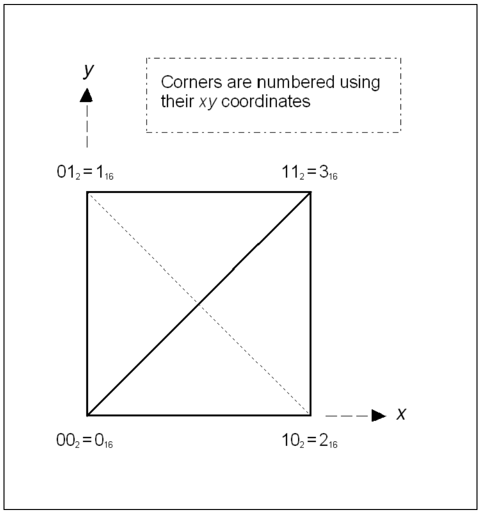

Figure 1. The two dimensional “volume,” a plane.

> **Axiom #2: All edges and diagonals will be denoted by the pair of vertex numbers of their ends, and the smaller vertex number will always be listed on the right.**

The distinguishing difference between an edge and a diagonal is that for an edge only one binary digit of the two vertices can differ. For example, $10_2 : 00_2$, $01_2 : 00_2$, $11_2 : 01_2$, and $11_2 : 10_2$ are edges, and $11_2 : 00_2$, and $01_2 : 10_2$ are diagonals.

> **Axiom #3: The edges of a unit volume are vertex pairs having only one binary digit of their vertex numbers that is different. All other vertex pairs are diagonals.**

The square has only two diagonals, but unit volumes of higher dimension have many more. For example, a three-dimensional unit cube has sixteen. It is our intent to tessellate each unit volume, regardless of the number of dimensions of its space, in such a way that all tessellated polyhedrons share a common diagonal as one of their edges, and the common diagonal chosen is the one connecting the vertices having the smallest and largest vertex numbers. This diagonal will be called the *principal diagonal*. If a unit volume is subdivided in this manner, all of its tessellated polyhedrons will be congruent. In the two-dimensional case the principal diagonal is $11_2 : 00_2$.

> **Axiom #4: The principal diagonal of a unit volume has as its ends the smallest and largest vertex numbers. All tessellated parts of the a unit volume will share the principal diagonal as one of their edges.**

The next construct needed is that of an exterior face of the unit volume. Faces are defined by two parallel edges, and edges are parallel if, and only if, a single common bit in each of their vertex numbers is different. Thus, the two edges $10_2 : 00_2$ and $11_2 : 01_2$ are parallel and the two edges $01_2 : 00_2$ and $11_2 : 10_2$ are parallel; the two edges of each pair differ by the requisite single common digit in each of their vertex numbers. But it is obvious from inspection of Figure 1 that both pairs of edges define the same exterior face (in the two-dimensional case, the only face -- the plane itself.) The two designations of the same face will certainly have the same diagonals. A simpler axiom can be used to eliminate the duplication. If faces are written having the vertices of their defining parallel edges placed in descending order, left to right, it is obvious that a common face designation $11_2 : 10_2 : 01_2 : 00_2$ results from both, and one of the pairs can be discarded as redundant. Formally,

> **Axiom #5: An exterior face is defined by two parallel edges.**
>
> **Axiom #6: Edges are parallel if, and only if, a single common bit in each of their vertex numbers is different.**

Axiom #7: External faces are represented by their four vertex numbers, two from each of their two defining parallel edges, placed in left-to-right descending order.

All that remains in is to divide each external face into two "half-faces," which are, of course, triangles. Because of the disciplined manner in which the vertices of the faces have been ordered, the three vertices of the half-faces can be produced using the leftmost two vertices with the rightmost vertex and the leftmost vertex with the rightmost two vertices. Thus, for the two-dimensional case, the two half-faces are $11_2 : 10_2 : 00_2$ and $11_2 : 01_2 : 00_2$. Note that this selection always produces half-faces that share the diagonal having the lowest and greatest vertex numbers as its ends. This is an important consideration. It insures that when tessellated unit volumes are abutted one to another, the touching faces of the two abutted volumes will have half-faces that each touch only one abutting half-face. Formally,

Axiom #8: The three vertices of the two half-faces produced from an external face are represented by (1) the leftmost two vertices combined with the rightmost vertex of the external face and (2) the leftmost vertex combined with the rightmost two vertices of the external face.

Determining of the number of trihedrons in the two-dimensional volume is trivial; there are two. However, determining the number of polyhedrons in high-dimension spaces is not. Axioms for determining that number and for determining their vertices are very important and will be discussed later.

Linear interpolation in the two-dimensional volume (plane) begins by selecting the trihedron in which the point of interest lies. If it lies on the $11_2 : 00_2$ diagonal, either trihedron will do, *since both will produce the same numeric result*. The criterion for selection is obvious: if $x \geq y$, the lower-right trihedron is chosen because it will completely enclose the point of interest; otherwise, if $x < y$, the upper-left trihedron is chosen. Selecting which polyhedron to use in higher dimensional cases will not be obvious. It will be necessary to have an efficient way of determining which polyhedron surrounds the point of interest. A construct called a *selector* that is based on the relative relationships of the coordinates of the point of interest (as in the two-dimensional case, where it is based on $x \geq y$ or $x < y$) will be defined for the higher dimensional cases later.

Interpolation within the trihedron, or triangle, can be defined in terms of the *natural coordinates* of the point of interest. The natural coordinates of a point within or on the boundaries of a trihedron are called "natural" because they correspond to the fractional distance of the point along a line from one side of the trihedron passing through the point and extending to the opposite vertex. Natural coordinated have a very useful property in that they all lie in the domain (0, 1) for all points within or on the boundary of the trihedron, and at least one of the coordinates is negative for points lying outside the trihedron. In other words, all of the natural coordinates will be the in domain (0, 1) for every interpolated point, and at least one will not be in the domain for every extrapolated point.

Interpolation within a trihedron can be defined in terms of the natural coordinates, a_i, of a *point of interest* as

$$f(x,y) = a_0 f(x_0,y_0) + a_1 f(x_1,y_1) + a_2 f(x_2,y_2), \qquad \text{where} \quad \sum_{i=0}^{2} a_i = 1.$$

The rectangular coordinates of the vertices are denoted as x_i, y_i. The dependent variables at the vertices are denoted as the functions $f(x_i, y_i)$. A further side condition is that the natural coordinates must sum to one. Expressing this relationship in matrix form,

$$[f(x,y)] = \begin{bmatrix} f(x_0,y_0) & f(x_1,y_1) & f(x_2,y_2) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

Based on the same natural coordinates, the coordinates of the point of interest can be written in terms of the rectangular coordinates of the vertices, as

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

It is now possible to solve for the natural coordinates in terms of the rectangular coordinates of the vertices, as

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Substituting for the natural coordinated, the interpolation equation can be written as

$$[f(x,y)] = \begin{bmatrix} f(x_0,y_0) & f(x_1,y_1) & f(x_2,y_2) \end{bmatrix} \begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Let us now consider a typical interpolation. The independent values *x* and *y* will be 0.6 and 0.7, respectively. Because *x* is less than *y*, the selector designates the upper-left trihedron having vertices 11_2, 01_2 and 00_2. The natural coordinates are computed as:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.6 \\ 0.7 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0.6 \\ 0.7 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.1 \\ 0.3 \end{bmatrix}$$

All of the natural coordinates are positive and in the domain (0, 1), as expected. Had the lower-right trihedron been mistakenly selected, the matrix of natural coefficients would have been $[0.7, -0.1, 0.4]^T$, with the negative coefficient showing required extrapolation outside the lower-right trihedron.

It is worth noting in passing that since the volume of interest will always be a unit volume having edges of unit length, the coordinates of its vertices will always be zeros or ones. Because of that, the inverse of the matrix of coordinates can have only values of 1, 0 , and -1 for its elements. **Thus, the natural coordinates can always be computed without multiplication, using only addition and subtraction.**

3. INTERPOLATING IN A THREE-DIMENSIONAL SPACE

All of the axioms developed for the two-dimensional case apply to the three-dimensional case. This is the nature of *axiom based induction*, and the axioms must apply not only to three- but also to four-, five-, six-, and seven-, and *n*-dimensional spaces as well. In three-dimensional space, as in two-, it is possible to produce a physical model of the volume. For two dimensions, the “volume” was approximated by a sheet of paper; for three dimensions, the cut-off end of a 4” by 4” fence post serves as approximation of the volume. It is possible, by applying duct tape between successive cuts, to saw a 4” cube of wood along three diagonals to produce six tetrahedrons. It is obvious that the six tetrahedrons completely fill the volume, and it is important that they do so. If any *n*-dimensional space is not filled by the polyhedrons into which it is tessellated, then points lying in the unfilled parts of the volume can not be interpolated. We can not consider algorithms that do not meet this criterion. Tetrahedrons that share a face must share the entire face. This criterion will cause interpolation across the shared face to be continuous. Kanamori and Kotera[5] have reported tessellation of a unit cube into five rather than six tetrahedrons. The method described here notes this fact in passing, but will develop the axioms for tessellation based on the six tetrahedrons that share the common principal diagonal $111_2 : 000_2$ (or $7_{16} : 0_{16}$). By Axiom #1, the number of vertices of the cube is eight, which is 2^3.

Axiom #3 defines how edges are to be produced, and how they are separated from diagonals. When applied to vertex 000_2 in three-dimensional space, the following edge progression is produced: 001_2:000_2, 010_2:000_2 and 100_2:000_2. The number of edges that can radiate from any vertex is *n*, the dimension of the space. But by this progression duplicate edges are produced; e.g., vertices 000_2 and 001_2 both produce the same edge 001_2:000_2. This leads to an expected value that can be used later for partial verification.

> **Expected Value #1: The number of edges of an n-dimensional volume is equal to the product of number of vertices times the dimensions of the space divided by two.**

Expected Values developed in defining the *axiom based* method are used as sign posts for consistent and expected behavior in the high-dimensional spaces where no "cut off ends of fence posts" can be used as a reality check. For three-dimensional space, the number of edges is twelve, a number that can be verified from a physical cube. For two-dimensional space, the number is 4, as expected. We can see consistency in two-dimensional, three-dimensional, and even one-dimensional spaces. It is reasonable to expect this consistency will hold for *n*-dimensional space as well.

Faces of the volume are defined by Axiom #5 and Axiom #6. For any given edge, such as edge 111_2:110_2, there exists the following progression of parallel edges that form faces: 011_2:010_2, 101_2:100_2 , and 110_2:111_2. But of the parallel edges formed following the axioms, one is a duplicate of the original (110_2:111_2 and 111_2:110_2). Therefore, we can conclude that each edge is a part of *n*-1 faces. But for every face formed in this manner there are three duplicate faces produced, one from each of the other edges of the face. Thus,

> **Expected Value #2: The number of external faces of an n-dimensional volume is equal to the product of number of edges times one less than the number of dimensions of the space divided by four.**

For three-dimensional space the number of faces is six, a number that can be verified from a physical cube. For two-dimensional space the number is 1, as expected.

3.1 Developing tetrahedrons by edge-matching

Half-faces associated with each face are formed according to the guidance of Axioms #7 and #8. Using only the list of these carefully constructed half-faces it is possible to determine the vertices of the tessellated polyhedrons that fill a unit volume in *n*-dimensional space. It should be noted that there is no prescribed order for the half-faces in the list.

For three-dimensional spaces, the polyhedrons are tetrahedrons, each having four triangular sides. The edge-matching algorithm begins with an initial search of the list of half-faces to find each instance of a half-face whose rightmost vertex is zero. For example, of the twelve half-faces in three-dimensional space, one meeting this criterion has vertices 101_2 : 100_2 : 000_2. The leftmost edge of the example half-face has vertices 101_2 : 100_2 . For each half-face found in the initial search that has a rightmost vertex of zero, the list of half-faces is searched a second time. The second search of the list is to find every half-face whose rightmost edge vertices match the leftmost edge vertices of the half-face found by the first search; in the example, the edge 101_2 : 100_2 . In the three-dimensional case, only one half-face that matches will be found on the second search, and it has vertices 111_2 : 101_2 : 100_2 . The leftmost vertex of the half-face found on the second search is concatenated to the left end of the original vertices, forming 111_2 : 101_2 : 100_2 : 000_2, one of the tetrahedrons of the unit cube. Vertices of the remaining five tetrahedrons are found in like manor, and are shown in Table 1.

The Six Tetrahedrons of Three-dimensional Space

Index	Tetrahedron Vertices
1	111_2 : 110_2 : 100_2 : 000_2
2	111_2 : 110_2 : 010_2 : 000_2
3	111_2 : 101_2 : 100_2 : 000_2
4	111_2 : 101_2 : 001_2 : 000_2
5	111_2 : 011_2 : 010_2 : 000_2
6	111_2 : 011_2 : 001_2 : 000_2

Table 1.

In the manner just described, each subsequent search of the list of half-faces finds at least one half-face with a matching edge. In three-dimensional case, only one subsequent search is required, but in the general case, the original search for half-faces with a vertex of zero is followed by *n*-2 subsequent nested searches of the list to find all matching edges. With each match found, one vertex is added to the left end of the sequence of polyhedral vertices, and the edge formed by that added vertex and the vertex to its right form the next edge to be searched for in the next nested search of the list of half-faces. This will be shown by example later in the discussion of four-dimensional space.

3.2 Deriving Tetrahedron Selectors

An efficient means is needed for selecting which of the six tetrahedrons is to be used for interpolation for any given point of interest; in other words, which tetrahedrons encloses the point of interest. It will be shown that tetrahedron *selectors* can be defined for this purpose, and their definition depends solely on the coordinates of the vertices of the six tetrahedrons. Once determined, the selectors to be used for a given point of interest will be determined by *conditions* developed solely from the coordinates of that point of interest.

Derivation of the selectors proceeds as follows. The vertices of each tetrahedrons are decomposed into a table shown in Table 2. The example tetrahedron used is #3, and its vertices are $111_2 : 101_2 : 100_2 : 000_2$. Each of the vertices of the tetrahedron is decomposed, right to left. The rightmost binary digit is associated with the coordinate *z*, the middle digit with *y*, and the leftmost with *x*. If the coordinates are compared two at a time, a logical True:False table can be built, as in Table 2. If, for example, $y < x$ for any one of the vertices, *y* will be less than or equal to *x* for all the vertices, and, more importantly, for all points within the tetrahedron defined by those vertices. It is impossible for *y* to be both less than *x* and greater than *x* in the same tetrahedron (or on its edges). It should be noted that if the two coordinates of a vertex have the same value, no conclusion can be drawn; hence the question marks in the table. A unique True:False table is constructed in like manner for each of the six tetrahedrons.

True:False Table for Tetrahedron #3

Vertex	*z*	*y*	*x*	$z<y$	$z<x$	$y<x$
000_2	0	0	0	?	?	?
100_2	0	0	1	?	True	True
101_2	1	0	1	False	?	True
111_2	1	1	1	?	?	?

Table 2.

A table of *selectors* can now be built from the True:False tables of the six tetrahedrons. The selector values have six binary digits, with each digit corresponding to one of the six tetrahedrons. The tetrahedrons are numbered right to left. A binary 1 represents the condition “True” and a 0 represents “False”. An example table of selectors is shown in Table 3. The “True” and “False” values for the example tetrahedron #3 are shown in bold face.

Tetrahedron *Selectors*

Condition	*Selector*
	654321
$z<y$	010**0**11$_2$
$z<x$	000**1**11$_2$
$y<x$	001**1**01$_2$
$z \geq y$	101**1**00$_2$
$z \geq x$	111**0**00$_2$
$y \geq x$	110**0**10$_2$

Table 3.

The tetrahedron selection process is thus dependent *only* on the coordinates of the point of interest in the unit volume. The selection is done by first determining the *conditions*, based on the coordinates, and then combining the selectors for those conditions using a logical “AND” operation. For example, if the [*x*,*y*,*z*] coordinates of the point of interest are [0.68, 0.53, 0.91], the conditions are $z \geq y$, $z \geq x$, and $y<x$ and the selectors are 101100_2, 111000_2, and 001101_2. The “AND” of the three selectors is 001000_2, designating tetrahedron #4 as surrounding the point of interest and the tetrahedron to be used for interpolation. This can be verified using the three-dimensional form for determination of the natural coordinates, and using the vertices of tetrahedron #4, which are $111_2 : 101_2 : 001_2 : 000_2$. Restating the equation for three dimensions:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.68 \\ 0.53 \\ 0.91 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.53 \\ 0.15 \\ 0.23 \\ 0.09 \end{bmatrix}$$

Selection of the correct tetrahedron is verified by the fact that all four of the natural coordinates are in the range [0,1].

Additional expected values can be derived from the selectors.

> **Expected Value #3: The number of selectors is the two times the sum of integers 1 to *n*-1, for *n* greater than two, and is two for *n* equal to two.**

The number of binary digits in each selectors is the same as the number of polyhedrons in the *n*-dimensional space, and is a more ominous number. Since all combinations of selectors are allowable, the number of polyhedrons in the unit volume must be *n*!.

> **Expected Value #4: The number of binary digits in each polyhedron selector for an *n*-dimensional space is equal to the number of polyhedrons filling a unit volume in that n-dimensional space, and is n!.**

The exploding number, *n*!, will probably dictate the early end of computational practicality. For a seven-dimensional space, the number of octahedrons needed to fill the space will be 7!, or 5040, and each selector will therefore have to be 5040 binary digits in length.

4. INTERPOLATING IN A FOUR-DIMENSIONAL HYPERSPACE

Equipped with the axioms stated above, and with the expected values defined as guideposts, we are ready to step off into a four-dimensional hyperspace where physical verification is impossible. From Axiom #1, the number of vertices of a four-dimensional volume will be sixteen. From Expected Values #1 and #2, the number of edges will be 32 and the number of external faces will be 24. From Expected Value #4, the number of polyhedrons expected is 4!, or 24, and the polyhedrons are pentahedrons, each having five triangular sides.

As with the three-dimensional space, half-faces associated with each face are formed according to the guidance of Axioms #7 and #8. As before, using only the list of half-faces, vertices of the tessellated pentahedrons that fill the unit volume can be determined. The edge-matching algorithm begins an initial search of the list of 48 half-faces to find each instance of a half-face whose rightmost vertex is zero. For example, one meeting this criterion has vertices $1010_2 : 1000_2 : 0000_2$, or $A_{16}:8_{16}:0_{16}$. The leftmost edge of that half-face has vertices $A_{16}:8_{16}$. An accumulating sequence of vertices that will ultimately contain the vertices of a pentahedrons is initialized with the three vertices from the first found half-face, and is $A_{16}:8_{16}:0_{16}$. For each half-face found on the initial search that has a rightmost vertex of zero, the list of half-faces is searched a second time. The second search of the list is to find every half-face whose rightmost edge vertices match the leftmost edge vertices of the half-face found by the initial search; for example, the edge $A_{16}:8_{16}$. In the four-dimensional case, two half-face will be found that match; they are $B_{16}:A_{16}:8_{16}$ and $E_{16}:A_{16}:8_{16}$. For the first half-face found on the second search of the list, the leftmost vertex is concatenated to the left of end of the accumulating vertex sequence, and the sequence becomes $B_{16}:A_{16}:8_{16}:0_{16}$. The leftmost two vertices of the sequence form yet another new edge, $B_{16}:A_{16}$. A third search of the list is started to find every half-face whose rightmost edge vertices match the leftmost edge vertices of the accumulating sequence of vertices found by the initial and second searches; for example, the edge $B_{16}:A_{16}$. The third search will find only one matching half face, $F_{16}:B_{16}:A_{16}$. The leftmost vertex is concatenated onto the left end of the accumulating sequence, forming $F_{16}:B_{16}:A_{16}:8_{16}:0_{16}$, which is one of the 24 pentahedrons of the four-dimensional hyperspace. Repeating the third search using the second half-face found on the second search produces $F_{16}:E_{16}:A_{16}:8_{16}:0_{16}$, another of the 24 pentahedrons of the four-dimensional hyperspace. The edge matching process beginning with an initial search and two nested subsequent searches of the list of half faces produces all of the expected 24 pentahedrons. The results are listed in Table 4.

The Twenty-four Pentahedrons of Four-dimensional Space

Index	Pentahedron Vertices	Index	Pentahedron Vertices	Index	Pentahedron Vertices
1	$F_{16}:E_{16}:C_{16}:8_{16}:0_{16}$	9	$F_{16}:D_{16}:9_{16}:8_{16}:0_{16}$	17	$F_{16}:D_{16}:5_{16}:4_{16}:0_{16}$
2	$F_{16}:D_{16}:C_{16}:8_{16}:0_{16}$	10	$F_{16}:B_{16}:9_{16}:8_{16}:0_{16}$	18	$F_{16}:7_{16}:5_{16}:4_{16}:0_{16}$
3	$F_{16}:E_{16}:C_{16}:4_{16}:0_{16}$	11	$F_{16}:D_{16}:9_{16}:1_{16}:0_{16}$	19	$F_{16}:D_{16}:5_{16}:1_{16}:0_{16}$
4	$F_{16}:D_{16}:C_{16}:4_{16}:0_{16}$	12	$F_{16}:B_{16}:9_{16}:1_{16}:0_{16}$	20	$F_{16}:7_{16}:5_{16}:1_{16}:0_{16}$
5	$F_{16}:E_{16}:A_{16}:8_{16}:0_{16}$	13	$F_{16}:E_{16}:6_{16}:4_{16}:0_{16}$	21	$F_{16}:B_{16}:3_{16}:2_{16}:0_{16}$
6	$F_{16}:B_{16}:A_{16}:8_{16}:0_{16}$	14	$F_{16}:7_{16}:6_{16}:4_{16}:0_{16}$	22	$F_{16}:7_{16}:3_{16}:2_{16}:0_{16}$
7	$F_{16}:E_{16}:A_{16}:2_{16}:0_{16}$	15	$F_{16}:E_{16}:6_{16}:2_{16}:0_{16}$	23	$F_{16}:B_{16}:3_{16}:1_{16}:0_{16}$
8	$F_{16}:B_{16}:A_{16}:2_{16}:0_{16}$	16	$F_{16}:7_{16}:6_{16}:2_{16}:0_{16}$	24	$F_{16}:7_{16}:3_{16}:1_{16}:0_{16}$

Table 4.

In the manner just described, each nested search of the list of half-faces finds at least one half-face with a matching edge. In general, the original search for half-faces having a vertex of zero is followed by *n*-2 subsequent nested searches of the list to find all matching edges. With each match found, one vertex is added to the left end of the accumulating sequence of polyhedral vertices, and the edge formed by that added vertex and the vertex to its right form the next edge to be searched for in the next nested search of the list of half-faces. Note that on the last search of the list, the final vertex added to the accumulating sequence of vertices always will have the value 2^n - 1. This process yields all tessellated polyhedrons of an *n*-dimensional space, and is the final axiom.

> **Axiom #9: An initial search of the list of half-faces is begun to find every half-face having a rightmost vertex of zero. For each such half-face found, an accumulating vertex sequence is initialized with the three vertices of that half-face, and *n* - 2 additional nested searches of the list of half-faces are performed. On each nested search, when a matching edge composed of the two rightmost vertices of a half-face from the list match the two leftmost vertices of the accumulating vertex sequence, the leftmost vertex of that half-face is concatenated to the left end of the accumulating sequence, and the next level of the nested search is begun. The vertex added to the accumulating vertex sequence on the most interior of the nested searches will always have the value (2^n - 1). The process is terminated when the initial search has examined all half-faces in the list.**

4.1 The pentahedron selectors

Derivation of the pentahedron selectors is analogous to the that for the tetrahedron selectors. There are twelve selectors, as expected, and each one is twenty-four digits in length. The selectors are shown in Table 5.

Pentahedron *Selectors*

Condition	*Selector* 24, …, 1	Condition	*Selector* 24, …, 1
$z<y$	$30F0F5_{16}$	$z \geq y$	$CF0F0A_{16}$
$z<x$	$03F05F_{16}$	$z \geq x$	$FC0FA0_{16}$
$z<w$	$0053FF_{16}$	$z \geq w$	$FFAC00_{16}$
$y<x$	$0F350F_{16}$	$y \geq x$	$F0CAF0_{16}$
$y<w$	$050F3F_{16}$	$y \geq w$	$FAF0C0_{16}$
$x<w$	$500FF3_{16}$	$x \geq w$	$AFF00C_{16}$

Table 5.

5. AN EXAMPLE APPLICATION: CONVERTING 3-DIMENSIONAL COLORS INTO 4-DIMENSIONAL TONER DENSITIES AND BACK

An application of the method of *measurement and interpolation* is demonstrated using the calibrated colors of a high-end **IBM Infoprint Color Printer**. The calibration processes, as implemented, requires use of both three-dimensional and four-dimensional interpolation methods. As such, the calibration process is an excellent vehicle for quantifying the magnitude of color errors resulting from the numerical interpolations used in its embodiment. The calibration is verified by choosing a set of 60 diverse colors, specified by their CIELAB L*a*b* coordinates, converting them to the primary toner densities required by the printer, and then reconverting the toner densities back to L*a*b* coordinates. This "round trip" requires two color-space conversions and allows numerical measurement of color errors between the initial and final L*a*b* color components.

The **IBM Infoprint Color Printer** uses four toners, Cyan, Magenta, Yellow and Black (CMYK). Its calibration entails producing artificial test images containing CMYK color squares. The color squares must be created using identical halftoning methods intended for subsequent color printing. The test images are then printed and the color of each printed square is measured. Nine density levels of each of the four toners were used for this example, the least level of each always being 0% and the greatest always 100%. Thus 9^4, or 6561, color squares were produced. Additionally, each color square was randomly placed in four different positions, thereby increasing the number of color squares printed to $4x9^4$, or 26,244. The color squares were measured using a GRETAG SpectroScan,™ and the four randomly positioned samples of each color were averaged to produce the final color measurements. The density variations, although not uniformly spaced for each toner, were applied regularly across the entire four-dimensional color volume producing regularly spaced tables of samples. The regularly spaced independent variables of each of the three table produced are the values of the CMYK densities used to produce the test square, and the dependent values in the three tables are the corresponding measured color components L*, a* or b*.

The regularly spaced grids of CMYK values are useful for producing all intermediate values of L*, a* and b* by four-dimensional interpolation, a separate interpolation for each of the three components. By using these interpolations tables, and with the formulae derived above, it is possible with iterative methods to produce a set of four tables that have as their independent variables uniformly spaced values of L*, a* and b*, with corresponding dependent values C, M, Y and K. This reuse of the measured data is frequently referred to as a "corner turn." The second set of tables, uniformly spaced in L*, a* and b*, are usable for the interpolation of CMYK values (requiring a separate three-dimensional interpolations for each C, M, Y, or K) that will produce each intermediate L*a*b* color. It is noted in passing that in the corner turn the toner density selections for each particular L*a*b* combination are frequently not unique. That is to say, there are often many distinct combinations of C, M, Y, and K that produce the same L*, a* and b*. Thus, a side condition is needed to select which of the many combinations is to be included in the four new tables. Two frequently used side conditions are 1) to choose the combination of CMYK that consume the least total of the four toners, or 2) the combination of CMYK that has the least Black component. Either choice produces four tables that have a single combination of values of CMYK for a particular combination of L*a*b*.

With the two sets of tables as just defined, the evaluation of color errors produced by two sequential color-space conversion processes can be evaluated. To do so, a first digitized image is artificially produces that has 60 diverse colors, specified by their L*a*b* coordinates. That first image is then converted into a printer-ready second image having equivalent pixels with CMYK color components. The values of the four CMYK components of each pixel in the second image are determined by interpolation using the L*a*b* values of a corresponding pixel in the first image. This requires four three-dimensional interpolations for each pixel and uses the second set of interpolation tables (the "corner-turned" set). Finally, a third image is produced from the second. Pixel values in the third image have L*a*b* components that are produced by interpolation using their corresponding CMYK pixel values in the second image. This requires three four-dimensional interpolations using the first set of tables, those tables produced by direct measurement of the printed color test squares.

With the “round-trip” processes completed, the L*a*b* pixel values in the third image are compared to the corresponding L*a*b* pixel values in the first image, and the Euclidean distance between the two colors, called the CIELAB ΔE, is used to quantify their difference. Numerical values for the 60 chosen colors are shown in Table 6.

The CIELAB ΔE is a convenient measure of the approximate perceptual difference between the two colors, or, in this case, the color error produced by the two color-space conversions and the implicit numerical truncation of their 8-bit color component values. A CIELAB ΔE of 1.0 or less is generally conceded as being near a “just noticeable difference,” and values less than 5.0 are considered an adequate match for most practical printing purposes. (As a practical matter, CIELAB ΔE differences of 5.0 can occur when the same color square is printed on opposite edges of a page).

The values of CIELAB ΔE greater than 4.0 in Table 6. are the consequence of an entirely different phenomenon. It is very easy to select real colors that are out of the gamut of the printer; that is, colors that can not be produced by any possible combination of CMYK toners available to the printer. In selecting colors near those of the Macbeth ColorChecker,™ a number of out-of-gamut colors were inadvertently selected. By moving each of those colors along a line toward the L* axis while holding L* constant, thereby desaturating them, the out-of-gamut colors are brought back into the gamut of the printer. Two versions of the “corner-turned” second table were used. The first had equally spaced grid increments of (2, 4, 4) for L*, a* and b*. This produces very large tables. The second version had equally spaced grid increments of (5, 10, 10) for L*, a* and b*, thereby reducing the table sizes by a factor of more than fifteen. Note also that extrapolation of colors near the edge of the printer’s gamut was prohibited. In retrospect, this prohibition may have been too strict, since it causes erosion of the edges of the color gamut and thereby decreases its size for each sequential color-space conversion. The coarser grid did not produce significantly larger color errors, a testament to the efficacy of the interpolation methods, but its use did cause a few additional colors to be out of the printers gamut because of edge erosion.

6. CONCLUSIONS FOR INTERPOLATING IN *n*-DIMENSIONAL HYPERSPACES

With considerable confidence, the axioms that have been developed have lead to the derivations of highly efficient[6] linear interpolation in three-, four-, five-, six-, and seven-dimensional hyperspaces. As a matter of fact, the method extends to hyperspaces beyond seven dimensions, but computational practicality begins to become a severe limit. As stated above, the number of octahedrons needed to fill a seven-dimensional space is 5040. Since the number of polyhedrons increases as *n*!, a nine-dimensional hyperspace requires a probably impractical 453,600 decahedrons to fill the hyperspace, and making measurements from which a nine-dimensional interpolation could be applied could become a career opportunity of tedium. But, at least for seven-dimensional hyperspaces, which allow color conversion from seven-ink or seven-toner color-spaces to other color-spaces, the computational complexity is a practical possibility. An interesting application might be the conversion of a seven-dimensional color-space to a four dimensional color-space that would allow a seven-dimensional *master image* to be “re-purposed” for printing by a less costly four-color method. Another application might be the use of a seven-dimensional interpolation, applied iteratively to produce a “corner turn,” that could allow the direct mapping of three-dimensional color into seven-dimensional ink densities.

The reproduction of color by using a small number of pigments, toners or inks on paper remains important in a society which can still be called “paper based.” This is true even though color image displays, such as cathode ray tubes, have been available for more than fifty years. Color images printed on paper dominate the still image market because of their availability, convenience, ease of long-term storage, and cost.

Original Color			Round Trip Color-Fine Spacing				Round Trip Color-Coarse Spacing				Generic Color
L*	a*	b*	L*	a*	b*	ΔE	L*	a*	b*	ΔE	
	15	15	39.2	15.2	14.8	0.3	39.2	15.2	15.9	0.9	Dark Skin
39	10	10	39	10.0	10.3	0.8	39.0	10.1	10.4	0.4	
39	5	5	39.1	5.5	4.9	0.5	38.8	5.2	5.1	0.3	
68	16	18	68.0	16.2	18.0	0.2	68.1	15.6	17.8	0.5	Light Skin
68	11	12	68.1	11.0	12.1	0.1	68.3	10.9	11.8	0.4	
68	5	6	68.1	4.9	5.9	0.2	68.3	5.4	6.2	0.5	
50	-4	-22	50.0	-4.2	-22.1	0.2	50.5	-4.3	-22.5	0.8	Blue Sky
50	-2	-15	50.5	-2.5	-15.2	0.7	51.1	-3.3	-15.7	1.8	
50	-1	-7	51.1	-1.6	-7.6	1.4	51.1	-1.7	-7.5	1.4	
43	14	20	43.2	-14.4	20.2	0.5	44.5	-17.4	22.2	***4.3***	**Foliage**
43	10	14	42.9	-10.0	13.9	0.1	44.5	-10.7	14.6	1.8	
43	-5	7	43.1	-5.0	7.5	0.5	43.7	-5.2	7.3	0.8	
55	9	-25	55.3	9.0	-23.1	1.9	55.2	8.6	-20.2	***4.8***	Blue Flower
55	3	-17	55.1	3.1	-17.1	0.2	54.6	2.7	-17.4	0.6	
55	3	-8	55.0	3.1	-8.0	0.1	55.6	2.8	-7.8	0.7	
71	32	-1	70.9	-28.2	-1.2	***3.8***	71.1	-27.4	-1.5	***4.6***	Bluish Green
71	22	-1	71.3	-22.4	-0.9	0.5	71.3	-21.4	-1.0	0.7	
71	11	0	71.2	-10.8	-0.2	0.3	71.0	-10.9	0.2	0.2	
62	33	57	62.0	28.6	51.9	***6.7***	62.4	29.5	50.7	***7.2***	Orange
62	22	38	62.0	22.3	38.7	0.8	62.2	21.8	38.7	0.8	
62	11	19	62.5	11.3	20.2	1.3	62.0	10.6	18.7	0.5	
40	9	-42	40.1	4.9	-27.8	***14.8***	40.1	7.8	-21.4	***20.6***	Purplish Blue
40	6	-28	40.3	6.1	-27.9	0.3	40.0	5.0	-24.3	3.8	
40	3	-14	40.0	2.8	-14.1	0.2	40.7	2.4	-14.3	1.0	
52	47	16	52.0	47.5	16.2	0.5	52.1	47.2	15.8	0.3	Moderate Red
52	31	11	52.1	31.0	11.1	0.1	52.5	30.6	11.0	0.6	
52	16	5	52.1	16.1	5.0	0.1	52.1	16.7	5.2	0.7	
31	21	-21	30.7	14.5	-16.1	***8.1***	30.7	12.0	-12.6	***12.3***	Purple
31	14	-14	30.9	14.0	-14.1	0.1	31.2	12.1	-14.8	2.1	
31	7	-7	31.0	7.2	-7.4	0.4	31.0	7.6	-7.1	0.6	
71	24	56	71.2	-24.1	56.2	0.3	72.2	-25.0	53.9	2.6	Yellow Green
71	16	37	72.5	-15.9	37.9	1.8	72.2	-15.9	37.4	1.3	
71	-8	19	70.9	-8.3	19.1	0.3	71.1	-8.2	18.9	0.2	
72	20	66	72.1	19.8	64.0	2.0	71.9	20.3	64.5	1.5	Orange Yellow
72	14	44	72.1	14.1	44.1	0.2	72.1	13.8	44.6	0.6	
72	7	22	72.1	6.9	22.0	0.1	72.0	7.1	22.1	0.1	
29	18	-51	29.2	6.0	-17.4	***35.7***	29.7	6.8	-15.5	***37.2***	Blue
29	12	-34	29.2	6.0	-17.4	***17.7***	29.7	6.8	-15.5	***19.2***	
29	6	-17	29.2	5.5	-15.4	1.7	29.4	4.2	-13.2	***4.2***	
54	39	32	54.3	-38.9	32.2	0.4	54.5	-36.2	28.2	***4.7***	Green
54	26	22	53.8	-26.3	22.1	0.4	54.1	-26.6	22.3	0.7	
54	13	11	54.4	-13.9	10.5	1.1	53.8	-13.5	11.1	0.5	
43	54	26	43.0	45.9	22.1	***9.0***	42.8	45.9	25.8	***8.1***	Red
43	36	18	43.0	36.6	18.1	0.6	42.7	36.6	18.3	0.7	
43	18	9	43.5	17.9	8.8	0.5	43.3	16.9	8.5	1.2	
82	4	77	82.1	4.0	76.0	1.0	83.4	5.1	75.7	2.2	Yellow
82	3	51	82.5	3.1	51.3	0.6	83.2	3.9	51.7	1.7	
82	1	26	82.2	1.0	26.3	0.4	82.9	1.7	27.0	1.5	
52	48	-13	52.1	35.1	-8.8	***13.6***	51.7	28.2	-10.2	***20.0***	Magenta
52	32	-9	52.3	31.6	-9.1	0.5	52.0	30.9	-9.1	1.1	
52	18	-4	51.9	18.0	-4.3	0.3	52.4	17.7	-4.5	0.7	
50	26	-28	50.1	-26.1	-28.5	0.5	50.6	-26.4	-28.0	0.7	Cyan
50	17	-19	50.3	-16.9	-18.9	0.3	50.8	-16.9	-18.8	0.8	
50	-9	-9	50.5	-8.9	-9.3	0.6	50.1	-8.8	-8.7	0.4	
100	0	0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	White 0.05
80	0	0	80.0	0.0	0.3	0.3	80.0	0.0	0.3	0.3	Gray 0.23
66	0	0	66.0	0.0	0.0	0.0	66.0	0.1	0.0	0.1	Gray 0.44
51	-1	-1	51.0	-1.1	-0.7	0.3	51.5	-1.2	-0.6	0.7	Gray 0.70
36	0	-1	36.0	0.0	-1.0	0.0	36.0	0.2	-1.1	0.2	Gray 1.05
21	0	-1	23.2	0.0	1.9	***3.6***	23.0	0.4	1.8	***3.5***	Black 1.50

Table 6. The Original Colors and Round Trip Colors Using Both Fine and Coarsely Spaced Tables for L*a*b* to CMYK Interpolation. (Out of gamut color are shown in bold face italics type)

REFERENCES

1. D. L. Post, and C. S. Calhoon, “An Evaluation of Methods for Producing Desired Colors on CRT Monitors,” *Color Research and Application,* 14(4), August 1989, pp. 172-186.
2. A. Cazes, G. Braudaway, et al., “On the Color Calibration of Liquid Crystal Displays,” *Proceedings of the SPIE International Conference on Electronic Imaging*, San Jose, CA, January 24-29, 1999, Vol. 3636.
3. M. Senechal, “Which Tetrahedra Fill Space?,” *Mathematics Magazine*, Vol. 54, No. 5, Nov. 1981, p. 241.
4. H. P. Manning**, The Geometry of Four Dimensions,** The Mcmillan Co, 1914 (Dover edition, 1956).
5. Kanamori, K., Kotera, H., et al., “Color Correction Technique for Hard Copies by 4-Neighbors Interpolation Method,” **Journal of Imaging Science and Technology**, Vol. 36, No. 1, Jan/Feb 1992, pp. 73-80.
6. J. M. Kasson, S. I. Nin, W. Plouffe, and J. L. Hafner, "Performing color space conversions with three-dimensional linear interpolation," **Journal of Electronic Imaging,** Vol. 4, No. 3, July 1995, pp. 226-250.