Populating the Matrix Entries in lutAtoBType and lutAtoBType of Version 4 ICC Profiles

George Pawle, Eastman Kodak Company

Introduction

One of the improvements of the version 4 ICC profile specification is the addition of a set of constants to the matrix operation in the lutAtoBType and lutAtoBType. The spec describes a simple matrix operation; however, there are some subtleties to using the matrix properly. This note describes how to populate the entries of the matrix to achieve the expected results.

The Matrix Operation

The spec describes 12 coefficients, e1 to e12, which are used in a matrix operation. Using x1, x2, and x3 as inputs and y1, y2, and y3 as outputs, the operation is:

y1 = x1*e1 + x2*e2 + x3*e3 + e10
y2 = x1*e4 + x2*e5 + x3*e6 + e11
y3 = x1*e7 + x2*e8 + x3*e9 + e12

The inputs and outputs are defined to be values in the range 0.0 to 1.0. There are several encodings (e.g., those used for ICC Lab and ICC XYZ) that may be used as the inputs and outputs, but the CMM implementing the matrix operation does not perform conversions based upon any of these encodings. It simply applies the above equations to the inputs and passes the outputs to the next processing stage.

The Matrix Entries

The coefficients needed for a given operation will depend upon the equations for the operation, the range of the inputs, and the range of the outputs. The matrix operates on inputs in the range 0.0 to 1.0; therefore, you need to map the range of the inputs to the input range of the matrix. This mapping is substituted into the equations for the operation. Similarly, the outputs from the matrix are in the range 0.0 to 1.0 and need to be mapped to the range of the outputs. This mapping is also substituted into the equations for the operation. Note that these formulas are used to help determine the coefficient values and are NOT implemented by the CMM!

In general, mapping ranges is done by mapping the minimum of the first range to the minimum of the second range and the maximum of the first range to the maximum of the second range. Intermediate values are linearly spaced between the minimums and maximums. Using
v1 = value in range 1, v2 = value in range 2, min1 = minimum of range 1, max1 = maximum of range 1, min2 = minimum of range 1, max2 = maximum of range 2, the equation is

v2 = (v1 – min1)*(max2 – min2)/(max1 – min1) + min2

The matrix operates on mapped values, so you need to know the formulas for converting mapped input values to input values and mapped output values to output values. For the input side, the input to the matrix uses the range 0.0 to 1.0, so min1 = 0.0 and max1 = 1.0. Using iv = input value and miv = the mapped input value, the input mapping formula is:

$$iv = (miv - 0.0)*(max2 – min2)/(1-0) + min2$$
$$iv = miv*(max2 – min2) + min2$$

The output from the matrix is in the range 0.0 to 1.0, the same as for the input. Using ov = output value and mov = the mapped output value, the output mapping formula is

$$ov = mov*(max2-min2) + min2$$

Once the mapping formulas have been substituted into the operation equations, the input variables are separated to determine the coefficients. Each of these coefficient values is converted to an s15Fixed16Number to obtain the final value to be put into the matrix entry of the profile.

Example
Determine the matrix coefficients for performing the linear part of the PCS Lab-to-PCS XYZ conversion. Assume that the curveType preceding the matrix has an identity operation.

Inverting the equations in Annex A of the spec gives

$$fX = a/500 + (L + 16)/116$$
$$fY = (L + 16)/116$$
$$fZ = -b/200 + (L +16)/116$$

The matrix operates on mapped values, so the mapping formulas must be determined. The preceding curveTypes implement an identity operation, so the inputs to the matrix operation will have PCS Lab encodings. For L*, the range is 0.0 to 100.0, so max2 = 100.0 and min2 = 0.0. Using mL = mapped L*

$$L = mL * (100.0 – 0.0) + 0.0$$
$$= mL * 100$$

The PCS a* and b* ranges are –128 to +127, so max2 = 127 and min2 = –128. Using ma = mapped a and mb = mapped b

$$a = ma*(127–(–128) + (–128)$$
$$= ma * 255 – 128$$
$$b = mb*(127–(–128) + (–128)$$
$$= mb * 255 – 128$$

Substituting these equations into the operation equations gives

$$fX = (ma * 255 – 128)/500 + ((mL * 100) + 16)/116$$
$$fY = ((mL * 100) + 16)/116$$
$$fZ = -(mb * 255 – 128)/200 + ((mL * 100) +16)/116$$

fX, fY, and fZ are in the range 0.0 to 1.0 and do not correspond to any particular encoding. They may be left in this state, or some additional function may be applied to them. This is a choice left to the profile builder. Whichever choice is made, it will need to be coordinated with the calculation of the curveType that follows the matrix. In this example, no additional function is applied, so no output mapping equations need to be applied to the equations.

Separating variables gives

$$fX = (ma * 255 - 128)/500 + ((mL * 100) + 16)/116$$
$$= mL * (100/116) + ma * (255/500) + (-128/500 + 16/116)$$

$$fY = ((mL * 100) + 16)/116$$
$$= mL * (100/116) + (16 /116)$$

$$fZ = -(mb * 255 - 128)/200 + ((mL * 100) +16)/116$$
$$= mL * (100/116) + mb * (-255/200) + (128/200 + 16/116)$$

Referring back to the matrix equations, $x1 = mL$, $x2 = ma$, $x3 = mb$, $y1 = fX$, $y2 = fY$, and $y3 = fZ$. Substituting into the above equations gives

$$y1 = x1 * (100/116) + x2 * (255 / 500) + (-128/500 + 16/116)$$
$$y2 = x1 * (100/116) + (16/116)$$
$$y3 = x1 * (100/116) + x3 * (-255/200) + (128/200 + 16/116)$$

From this we see that the matrix coefficients are

e1 = 100/116
e2 = 255/500
e3 = 0
e4 = 100/116
e5 = 0
e6 = 0
e7 = 100/116
e8 = 0
e9 = -255/200
e10 = -128/500 + 16/116
e11 = 16/116
e12 = 128/200 + 16/116

These are all s15Fixed16Numbers in the profile format. Their encodings are generated by multiplying by 65536, rounding to the nearest integer, and converting to hexadecimal. The actual numbers in the profile are

e1 = DCB1h
e2 = 828Fh
e3 = 0
e4 = DCB1h
e5 = 0
e6 = 0
e7 = DCB1h
e8 = 0
e9 = FFFEB99Ah
e10 = FFFFE1C6
e11 = 234F
e12 = C726