# A solution to CIECAM02 numerical and range issues

***Graeme W. Gill, Computer Graphics Technology P.L., Melbourne Australia.***

## Abstract

*CIECAM02 [1] is being used increasingly in color management systems as a gamut mapping space. There continue to be issues with using the published transformation in practical settings, due to the need to gracefully handle non-real world colors. The approach presented here is to extend the behavior of CIECAM02 to be unrestricted in range, to fully round trip in either direction, and do so in a way that is consistent with its defined behavior for real world colors. The result is an implementation of CIECAM02 that can be used as a drop in replacement for traditional gamut mapping and clipped spaces such as L*a*b*.*

## Introduction

One of the uses that color appearance spaces may be put to, is as a space for gamut mapping and clipping. Gamut mapping can be regarded as a geometric process of mapping from one 3 dimensional space to another. While the nature of the mapping is often guided by the gamut limits of the two devices involved, the overall desired transformation range may be dictated by other factors, such as the table ranges of the L*a*b* cube specified by the ICC profile B2A table. Negative luminance values may arise when mapping from spaces that have radically different surround conditions or illuminant white points. Both these situation can result in non real world color coordinates, that ideally should be dealt with gracefully within the gamut mapping process. Unfortunately while CIECAM02 is well behaved for real world colors, is not well behaved in many regions outside this range.

These difficulties have lead to the suggestion of a few partial solutions, the main one being pre-clipping color values in another color space such as XYZ prior to conversion into CIECAM02 space [2][3]. Drawbacks of this are that such a two step transformation is not easily reversible to form a round trip solution, and clipping in some other color space would seem to defeat much of the purpose of choosing CIECAM02 as a space to do gamut mapping and clipping in. Other suggestions [4] tackle some of the the round trip errors noticed in CIECAM02 over a limited range (i.e. +ve XYZ).

Additional shortcomings of CIECAM02 in regard to spectrum locus colors falling outside the Hunt-Pointer-Estevez (HPE) primary "triangle", and thereby falling afoul of the infinite slope at zero of the non-linear response compression has been addressed in [5] by a slight change to the HPE primaries. The issues of whether the intended CIECAM02 behavior is actually reasonable for some combinations of real world colors and adapted state raised in [5], are outside the scope of any approach that seeks to implement the CIECAM02 behavior in a manner consistent with its defined behavior for colors on and within the spectrum locus.

## Re-stating of the forward and reverse transformations

It is advantageous for the purposes of examining the nature of the numerical and range issues, as well as applying the modifications described in the next section, to re-state the basic CIECAM02 transformation from XYZ to Jab in a form slightly different (but mathematically equivalent) to the equations used in the original CIECAM02 paper [1].

Incoming XYZ values are transformed first by a 3x3 matrix into a sharpened cone space for chromatic adaptation, then transformed by another 3x3 matrix into the primary RGB cone space in the same manner as the standard CIECAM02.

Each RGB value is then transformed by a non-linear function into the Post-adapted cone response $R'_a, G'_a$ and $B'_a$. The numerical issues arising from this transformation will be dealt with latter in this paper. For now they are ignored, and the more serious underlying issues examined.

For the purposes of the discussion, the variables $e, nn, Nbb, c, z$ are regarded as constants, although in practice they depend on the viewing conditions. Some new variables are also introduced (*nn*, *ttA*, *ttd*, *S* and *J'*), to represent components of some of the equations in [1].

From [1](23), *nn* is a modified version of *n*:

$$nn = (1.64 - 0.29^n)^{0.73} \quad (1)$$

$R'_a, G'_a$ and $B'_a$ are then transformed into another 4 basis vectors:

a & b: Preliminary red-green & yellow-blue opponent coordinates from [1](14) and [1](15):

$$a = R'_a - (12/11) G'_a + (1/11) B'_a \quad (2)$$

$$b = (1/9) R'_a + (1/9) G'_a - (2/9) B'_a \quad (3)$$

ttA: Preliminary Achromatic response from [1](20):

$$ttA = 2R'_a + G'_a + (1/20) B'_a \quad (4)$$

ttd: Preliminary saturation scaling denominator (part of [1](16)).

$$ttd = R'_a + G'_a + (21/20) B'_a \quad (5)$$

The preliminary achromatic response *ttA* is transformed by a non-linear function into the Lightness value (remainder of [1](20)), and the lightness value *J* is broken up into a scale 1 value *J'*, and the original scale 100 value *J*:

$$A=(ttA-0.305)Nbb \quad (6)$$

$$J'=\left(\frac{A}{A_w}\right)^{cz} \quad (7)$$

$$J=J'\cdot 100 \quad (8)$$

a, b are transformed into the final values by multiplying by a scaling factor. The extraction of this as a separate variable $ss$ is the primary aim of the re-arrangement of the original equations:

$$S=\sqrt{a^2+b^2} \quad \text{(from [1](16))} \quad (9)$$

$$ss=\frac{\sqrt{J'}\,nn\,e^{0.9}}{ttd^{0.9}\,S^{0.1}} \quad (10)$$

$$a_c=a\cdot ss \quad (11)$$

$$b_c=b\cdot ss \quad (12)$$

$$C=\sqrt{a_c^2+b_c^2} \quad (13)$$

The values $J$, $a_c$ and $b_c$ being the desired gamut mapping space abbreviated as "Jab". The equivalent equation for inverting the transformation needs to recover $ss$ using just the output values is:

$$ss=\frac{J'^{(1.0/1.8)}nn^{(1.0/0.9)}e}{ttd\,C^{(1/9)}} \quad (14)$$

Equation (14) depends not only on information that is directly available when inverting the model (i.e. $J$, $a_c$ and $b_c$), but on a fourth unrelated vector *ttd* that was computed in the forward transform from the unscaled *a* & *b*. The use of a fourth vector *ttd* seems to be a feature unique to the CIECAM series of appearance models, and is at the root of many of its practical numerical difficulties. To invert the model, *ttd* needs to be computed from $J$, $a_c$ and $b_c$.

This can be done by noting that *ttd* can be computed from weightings of *ttA*, *a* & *b*:

$$ttd=w1\cdot ttA+w2\cdot a+w3\cdot b \quad (15)$$

$$w1=1.0 \quad (16)$$

$$w2=-11.0/23.0 \quad (17)$$

$$w3=-108.0/23.0 \quad (18)$$

and that

$$ss=\frac{1}{ttd\cdot ssp} \quad (19)$$

where

$$ssp=\frac{C^{(1/9)}}{e\,J^{(1.0/1.8)}nn^{(1.0/0.9)}} \quad (20)$$

$$a=a_c\cdot ss^{-1} \quad (21)$$

$$b=b_c\cdot ss^{-1} \quad (22)$$

substituting these into (15) gives:

$$ttd=w1\cdot ttA+ttd\cdot ssp(w2\cdot a_c+w3\cdot b_c) \quad (23)$$

and collecting ttd to the left hand side gives

$$ttd=\frac{w1\cdot ttA}{1-ssp(w2\cdot a_c+w2\cdot b_c)} \quad (24)$$

resulting in

$$ss=\frac{1-ssp(w2\cdot a_c+w2\cdot b_c)}{ssp\cdot w1\cdot ttA} \quad (25)$$

Once the scale factor $ss$ is known, *a* & *b* can be computed, leading on to the reconstruction of $R'_a$, $G'_a$ and $B'_a$ and ultimately XYZ.

## Numerical issues

Assuming the above equations are being used for the conversion, the core numerical issues boil down to those affecting the forward model scale factor *ss* (10).

At its most difficult, $J = 0$, $a = b = 0$ (implying $S = 0$) and $ttd = 0$ (It's assumed that if $ttA = 0$, then $J$ would have to be clipped to 0 to avoid taking a power of a negative number). So for instance

$$a_c=a\frac{\sqrt{J'}\,e^{0.9}nn}{ttd^{0.9}\,S^{0.1}} \quad (26)$$

becomes

$$a_c=0\frac{\sqrt{0}\,e^{0.9}\,nn}{0^{0.9}0^{0.1}} \quad (27)$$

As well as the denominator being the sources of two infinities, the zeros will cause infinities for the reverse conversion, and the behavior of the variables as they approach zero will have a dramatic effect on the sign and magnitude of the scale factor. Note also that *a* appears both on the numerator and disguised in *S* on the denominator.

Looking at the vectors *ttd* and *ttA* and their corresponding zero value planes in $R'_a$, $G'_a$ and $B'_a$ or $ttA$, $a$ and $b$ space, they must cross at a line. Near this crossing the scale factor undergoes large changes over a small range of input. The situation with regard to points lying near $a^2+b^2=0$ is similar, although the geometry is different, and of course all these effects interact near the zero point.

For the reverse calculation, the necessity of computing ttd from $a_c$ and $b_c$, adds another influential constraint, as it introduces another infinity if $ssp(w2\cdot a_c+w2\cdot b_c)=1$.

## Numerical solutions - Forward transform

We now go back to deal with the post-adaptation non-linearity response skipped over in the preliminary discussion. It causes two problems, one is that it asymptotes to a value of 400 in the forward direction for large input values, meaning that there is no reverse mapping for values over this. The original article recommended implementing a symmetric response for negative values, but this has the effect when used in the reverse direction of greatly exaggerating small negative CAM values. To mitigate these issues the curve is modified in a way that preserves it shape over a real world range, and adds a tangential linear extension for large positive values so that the range of positive values is not limited, and for negative input values a straight line extension is added that preserves negative values but expands them more than the positive characteristic on conversion to the CAM space, so that the values are compressed more on conversion from the CAM back to XYZ, helping to minimize shifts in resulting color for CAM values near or just below zero.

Lower extension:

$$xl_{\text{lim}}=0.005\times 100 \quad (28)$$

$$xl_{\text{val}}=\frac{400.0\,(FL\,xl_{\text{lim}}/100)^{0.42}}{27.13+(FL\,xl_{\text{lim}}/100)^{0.42}}+0.1 \quad (29)$$

$$xl_{\text{slope}}=(xl_{\text{val}}-0.1)/xl_{\text{lim}} \quad (30)$$

Upper extension:

$$xu_{\text{lim}}=100000\times 100 \quad (31)$$

$$xu_{\text{val}}=\frac{400.0\,(FL\,xu_{\text{lim}}/100)^{0.42}}{27.13+(FL\,xu_{\text{lim}}/100)^{0.42}}+0.1 \quad (32)$$

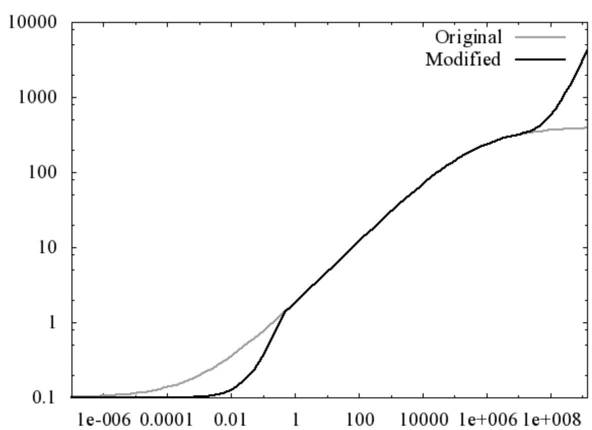
$$xu_{\text{slope}}=\frac{\partial}{\partial FL}\left(\frac{400.0\,(FL\,R'/100)^{0.42}}{27.13+(FL\,R'/100)^{0.42}}\right) at\ xu_{\text{lim}} \quad (33)$$

Non linearity equation:

$$\begin{aligned}&if\ R'<xl_{\text{lim}}\\&\quad R'_a=xl_{\text{slope}}R'+0.1\\&else\,if\ R'>xu_{\text{lim}}\\&\quad R'_a=xu_{\text{val}}+xu_{\text{slope}}(R'-xu_{\text{lim}})\\&else\end{aligned} \quad (34)$$

$$R'_a=\frac{400.0\,(FL\,R'/100)^{0.42}}{27.13+(FL\,R'/100)^{0.42}}+0.1 \quad (35)$$

Plot of modified non-linearity




Equation (9) is modified to create a version of $S$ restricted to be a minimum of a small positive value, to prevent it causing a divide by zero in the computation of $ss$. The exact minimum value isn't critical since the $a,b$ numerator dominates the limit as neutral colors are approached:

$$S_r=max\{\sqrt{a^2+b^2},1\times 10^{-12}\} \quad (36)$$

Equation (7) is modified to avoid taking the power of a negative number, and to switch from the power curve applied to A in computing $J'$, to straight line transfer curve that minimizes exaggerating the imaginary "blacker than black" colors:

$$J'_{\text{lim}}=0.005 \quad (37)$$

$$A_{\text{lim}}=Aw\,J'^{\,1.0/(cz)}_{\text{lim}} \quad (38)$$

$$\begin{aligned}&if\ A<A_{\text{lim}}\\&\quad J'=A\cdot J'_{\text{lim}}/A_{\text{lim}}\\&else\\&\quad J'=(A/A_w)^{cz}\end{aligned} \quad (39)$$

A constrained positive non-zero $J'$ is also computed for use in the scaling equation components:

$$J'_{c\text{lim}}=0.005 \quad (40)$$

$$\begin{aligned}&if\ A\le 0\\&\quad J'_c=J'_{c\text{lim}}\\&else\\&\quad J'_c=max\{(A/A_w)^{cz},J'_{c\text{lim}}\}\end{aligned} \quad (41)$$

The scaling factor equation (10) is divided into three components:

$$k1=\frac{nn^{1.0/0.9}\,e\,J'^{\,1.0/1.8}_c}{S_r^{1.0/9.0}} \quad (42)$$

$$k2=J'^{\,1/(CZ)}_c\frac{Aw}{Nbb}+0.305 \quad (43)$$

$$k3=w2\,a+w3\,b \quad (44)$$

The components combining thus:

$$ss^{(1/0.9)}=\frac{k1}{k2+k3} \quad (45)$$

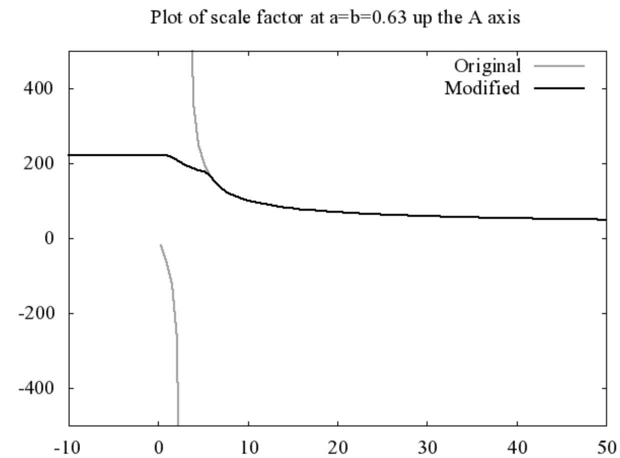
The use of $S_r$ and $J'_c$ in computing k1 and k2 ensures that the scale factor will not be zero, negative or infinity, due to them. To ensure that it does not go negative or to infinity due to k3, the ratio of k3 to k2 is constrained :

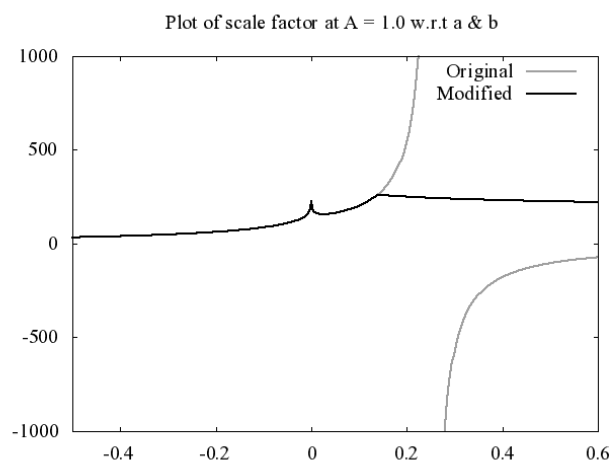
$$\begin{aligned}&ss_{llim}=0.55\\&if\ -k3>k2\cdot ss_{llim}\\&\quad k3=-k2\cdot ss_{llim}\end{aligned} \quad (46)$$

While this ensures good behavior in the forward transformation, the resulting values can still trigger problems in the reverse transformation, so an additional constraint needs to be applied. A preliminary scaling $ss_p$ factor is computed using equation (45) to see if this is the case, and if so the k3 value adjusted. (see also (53) and (54)).

$$\begin{aligned}&ss_{ulim}=0.9993\\&if\ ss_p\cdot k3>\frac{ss_{ulim}\cdot k1}{ss_p^{(1.0/9.0)}}\\&\quad ss=\left(\frac{k1\cdot(1-ss_{ulim})}{k2}\right)^{0.9}\\&else\\&\quad ss=ss_p\end{aligned} \quad (47)$$

The final $J$, $a_c$ and $b_c$ values are computed using equations (8), (11) and (12).




## Reverse transform

For the reverse transform, a similar, complementary procedure is used. First $C$ is computed using equation (13), and then a constrained version that has a minimum of a small positive value is computed. The exact minimum value isn't critical since $a, b$ numerator dominates the limit as neutral colors are approached.

$$C_r = max\{C, 1\times10^{-12}\} \tag{48}$$

The achromatic response $A$ is computed from $J'$ using the inverse of equation (39). The preliminary achromatic response $ttA$ is then computed using the inverse of equation (6). The constrained positive non-zero $J_c'$ is computed using equation (41).

The reverse scaling factor equation (25) is divided in a very similar fashion to the forward one, with one identical, and two similar components:

$$k1' = \frac{nn^{1.0/0.9}\, e\, J_c'^{\,1.0/1.8}}{C_r^{1.0/9.0}} \tag{49}$$

$$k2 = J_c'^{\,1/(CZ)} \frac{Aw}{Nbb} + 0.305 \tag{50}$$

$$k3' = w2\, a_c + w3\, b_c \tag{51}$$

The components combining to form the scale factor:

$$ss = \frac{(k1' - k3')}{k2} \tag{52}$$

Note that:

$$k1' = k1 / ss^{1.0/9.0} \tag{53}$$

$$k3' = k3 \cdot ss \tag{54}$$

The use of $J_c'$ and $C_r'$ in computing $k1'$ and $k2$ ensures that the scale factor will not be zero, negative or infinity, due to them. To ensure that $ss$ does not go negative or its inverse to infinity due to $k3'$, the ratio of $k3'$ to k1 is constrained :

$$\begin{aligned} &if\ k3' > k1' \cdot ss_{ulim} \\ &\quad k3' = k1' \cdot ss_{ulim} \end{aligned} \tag{55}$$

While this ensures good behavior in the reverse transformation, the resulting values can still trigger problems in the forward transformation, so an additional constraint needs to be applied. A preliminary scaling $ss_{pr}$ factor is computed using equation (52) to see if this is the case:

$$\begin{aligned} &if\ -k3' > ss_{pr} \cdot k2 \cdot ss_{llim} \\ &\quad ss = \frac{k1'}{k2 \cdot (1.0 - ss_{llim})} \\ &else \\ &\quad ss = ss_{pr} \end{aligned} \tag{56}$$

With $ss$ recovered, the inverses of equations (11) and (12) can be used to recover $a$ and $b$ from $a_c$ and $b_c$, and then $R_a'$, $G_a'$ and $B_a'$ are easily computed from $ttA$, $a$ and $b$ by inverting (2), (3) and (4).

The particular limit values chosen of 0.55 and 0.9993 were primarily arrived at through a process of numerical simulation, and strive to minimize the extremes of the scale factor while not impinging on the result for colors that lie on or inside the spectrum locus.

## Verification

Several numerical verifications were carried out. One was to compare the results for colors on and within the spectrum locus to the results from the transformation without the range limitation modifications, verifying that the results are unchanged within the numerical precision of the implementation. The other verifications are round trips:

Regularly sampling an XYZ cube in the range -100 to 120 for each coordinate, converting to Jab and then back to XYZ, with a worst case error of 7e-7.

Regularly sampling a Jab cube in the range J -50 to 115, a,b range -128 to +128, converting to XYZ and then back to Jab, with a worst case error of 4e-6.

The following two illustrations show a cylindrical grid in L*a*b* D50 color space of radius 50 and L* ranging from -100 to

+100, transformed to Jab space and plotted in 3D. The axes shown are from a,b -128 to +128, and J 0 to 100:

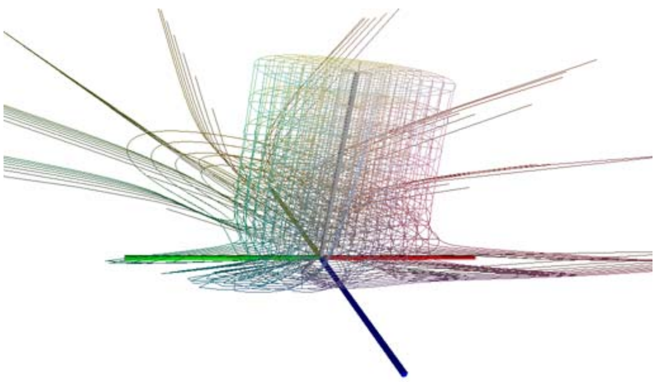


*Illustration 1: D50 L*a*b* cylindrical grid plotted in original CIECAM02 Jab space.*

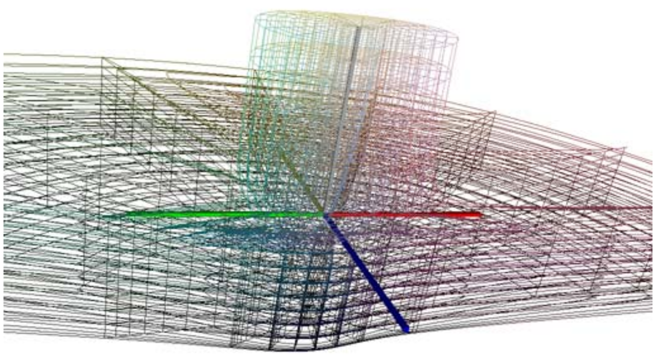


*Illustration 2: D50 L*a*b* cylindrical grid plotted in modified CIECAM02 Jab space.*

## Conclusion

The approach described here provides a robust solution for the numerical issues inherent in the CIECAM02 transform, at the cost of some extra complexity. It seems to be quite difficult to evaluate the geometry of color spaces in imaginary color regions and the effect this has on gamut mapping and clipping, because it does not seem possible to evaluate the subjective accuracy of such mappings through any practical experiment. It's possible to compare the behavior to some other color spaces (e.g. L*a*b* space) that are capable of representing imaginary colors, but since there is no basis to regard one or the other as canonical, it is hard to draw conclusions. Imaginary colors can inherently contain contradictions such as zero or negative luminance levels combined with very highly saturated colors, that make broad arguments about what's reasonable, confounding as well. Given these difficulties, it is probable that the particular way the scale factor has been broken down, and the choice of limiting values could well be improved in regard to how consistent end sensible the resulting gamut mapping space is. In addition, the nature of the CIECAM02 transformation (particularly the influence of (5)) causes what may seem to be surprising behavior compared to a space such as L*a*b*, since some Jab colors with quite negative J values transform into XYZ values that have very positive Y values, and hence will not be black. Whether this is of any consequence in applying gamut mapping to real world color transformations, and whether this behavior could be modified without deviating from CIECAM02 behavior within the real world color space is a topic for further research.

Perhaps the analysis in the early part of this paper of the existing CIECAM02 transformation in terms of the 4 basic vectors *a, b*, *ttA* & *ttd* and the scaling factor *ss* may inspire some simplification in future CAM's.

## Source Code

Source code for a CIECAM02 implementation that includes the modifications described in this paper is available under the GNU license as part of the ArgyllCMS package, from http://www.argyllcms.com, in the file xicc/cam02.c

## References


[1] Nathan Moroney, Mark D. Fairchild, R.W.G. Hunt, Changjun Li and M. Ronnier Luo and Todd Newman, "The CIECAM02 Color Appearance Model", *Proc. IS&T/SID 10th Color Imaging Conference*, pp. 23-27 (2002).

[2] Ingeborg Tastl, Miheer Bhachech, Nathan Moroney, and Jack Holm, "ICC Color Management and CIECAM02", Proc. IS&T/SID 13th Color Imaging Conference, pp. 217-223 (2005).

[3] Randy Guay and Mark Shaw, "Dealing with Imaginary Color Encodings in CIECAM02 in an ICC workflow", Proc. IS&T/SID 13th Color Imaging Conference, pp. 318 (2005).

[4] Chunghui Kuo, Eric Zeise, and Di Lai, "Robust CIECAM02 Implementation and Numerical Experiment within an ICC Workflow", Proc. IS&T/SID 14th Color Imaging Conference, pp. 215-219 (2006).

[5] Sabine Süsstrunk and Michael H. Brill, "The Nesting Instinct: Repairing Non-Nested Gamuts in CIECAM02", IS&T/SID 14th Color Imaging Conference, Late Breaking News Handout (2006).


## Author Biography


***Graeme W. Gill*** *is a independent consultant and software & hardware developer. He received a B.E. degree in Electronic Engineering in 1984 from the Royal Melbourne Institute of Technology, before working for Labtam P.L and Colorbus P.L over successive decades. Mr Gill is a member of the IEEE, ACM and IS&T.*