Algorithmic improvements for the CIECAM02 and CAM16 color appearence models

Nico Schlömer

February 14, 2018

This note is concerned with the CIECAM02 color appearance model and its successor, the CAM16 color appearance model. Several algorithmic flaws are pointed out and remedies are suggested. The resulting color model is algebraically equivalent to CIECAM02/CAM16, but shorter, more efficient, and works correctly for all edge cases.

1. Introduction

The CIECAM02 color appearance model [1] has attracted much attention and was generally thought of as a successor to the ever so popular CIELAB color model. However, it was quickly discovered that CIECAM02 breaks down for certain input values. A fair number of research articles suggests fixes for this behavior, most of them by modifying the first steps of the forward model. Luo and Li give an overview of the suggested improvements [2]; see references therein. Most recently, Li and Luo [3] gave their own suggestion on how to best circumvent the breakdown [3]. The updated algorithm differs from the original CIECAM02 only in the first steps of the forward model.

It appears that that the rest of the algorithm has not received much attention over the years. In both CIECAM02 and its updated version CAM16, some of the steps are more compli-cated than necessary, and in edge cases lead to break downs once again. The present document describes those flaws and suggests improvements (Section 2). The resulting model description (Section A) is entirely equivalent to the CIECAM02/CAM16, but is simpler – hence faster and easier to implement – and works in all edge cases.

All findings in this article are implemented in the open-source software package colorio [4].

2. Flaws and fixes

This section describes the flaws of CAM16 and suggests fixes for them. Some of them are trivial, others are harder to see. All listed steps also appear in the CIECAM02 color appearance model and trivially apply there.

2.1. Step 3, forward model

The original Step 3 of the forward model reads

Step 3. *Calculate the postadaptation cone response (resulting in dynamic range compression).*

$$R_a = 400 \frac{\left(\frac{F_L R_c}{100}\right)^{0.42}}{\left(\frac{F_L R_c}{100}\right)^{0.42} + 27.13} + 0.1$$

	with fixes	without
J	0.0	3.258e-22
C	0.0	4.071e-24
h	0.0	0.0
Q	0.0	2.233e-10
M	0.0	2.943e-24
s	0.0	1.148e-05

Table 1: CAM16 values upon input $X = Y = Z = 0$ with and without the fixes in this article. The exact solutions are zeros for every entry.

If R_c is negative, then

$$R_a = -400 \frac{\left(\frac{-F_L R_c}{100}\right)^{0.42}}{\left(\frac{-F_L R_c}{100}\right)^{0.42} + 27.13} + 0.1$$

and similarly for the computations of G_a and B_a.

If the sign operator is used here as it is used later in step 5 of the inverse model, the above description can be shortened.

Furthermore, the term 0.1 is added here, but in all of the following steps in which R_a is used – except the computation of t in Step 9 –, it cancels out algebraically. Unfortunately, said cancellation is not always exact when computed in floating point arithmetic. Luckily, the adverse effect of such rounding errors is rather limited here. The results will only be distorted for very small input values, e.g., $X = Y = Z = 0$; see Table 1. For the sake of consistency, it is advisable to include the term 0.1 only in the computation of t in Step 9:

$$R'_a = 400 \operatorname{sign}(R_c) \frac{\left(\frac{F_L |R_c|}{100}\right)^{0.42}}{\left(\frac{F_L |R_c|}{100}\right)^{0.42} + 27.13}.$$

2.2. Linear combinations, forward model

In the forward model, four linear combinations of R'_a, G'_a, and B'_a have to be formed. They can conveniently be expressed as the matrix-vector multiplication

$$\begin{pmatrix} p'_2 \\ a \\ b \\ u \end{pmatrix} := \begin{pmatrix} 2 & 1 & \frac{1}{20} \\ 1 & -\frac{12}{11} & \frac{1}{11} \\ \frac{1}{9} & \frac{1}{9} & -\frac{2}{9} \\ 1 & 1 & \frac{21}{20} \end{pmatrix} \begin{pmatrix} R'_a \\ G'_a \\ B'_a \end{pmatrix}$$

which on many platforms can be computed significantly faster than four individual dot-products. The last variable u is used in the computation of t in step 9.

2.3. Step 9, forward model

Step 9. *Calculate the correlates of [. . .] saturation (s).*

$$s := 100\sqrt{M/Q}.$$

This expression is not well-defined if $Q = 0$, a value occuring if the input values are $X = Y = Z = 0$. When making use of the definition of M and Q, one gets to an expression for s that is well-defined in all cases:

$$\alpha := t^{0.9}(1.64 - 0.29^n)^{0.73}, \qquad (1)$$

$$s := 50\sqrt{\frac{c\alpha}{A_w + 4}}.$$

2.4. Steps 2 and 3, inverse model

Step 2. *Calculate t, e_t, p_1, p_2, and p_3.*

$$t = \left(\frac{C}{\sqrt{\frac{J}{100}}(1.64 - 0.29^n)^{0.73}} \right)^{\frac{1}{0.9}},$$

$$e_t = \frac{1}{4}\left[\cos(h'\pi/180° + 2) + 3.8\right],$$

$$p_1 = \frac{50000}{13}N_c N_{cb} e_t \frac{1}{t},$$

$$p_2 = \frac{A}{N_{bb}} + 0.305,$$

$$p_3 = \frac{21}{20}.$$

Step 3. *Calculate a and b. If $t = 0$, then $a = b = 0$ and go to Step 4. In the next computations be sure transform h from degrees to radians before calculating $\sin(h)$ and $\cos(h)$:*
If $|\sin(h)| \geq |\cos(h)|$ then

$$p_4 = \frac{p_1}{\sin(h)},$$

$$b = \frac{p_2(2 + p_3)\frac{460}{1403}}{p_4 + (2 + p_3)\frac{220}{1403}\frac{\cos(h)}{\sin(h)} - \frac{27}{1403} + p_3\frac{6300}{1403}},$$

$$a = b\frac{\cos(h)}{\sin(h)}.$$

If $|\cos(h)| > |\sin(h)|$ then

$$p_5 = \frac{p_1}{\cos(h)},$$

$$a = \frac{p_2(2 + p_3)\frac{460}{1403}}{p_5 + (2 + p_3)\frac{220}{1403} - \left(\frac{27}{1403} - p_3\frac{6300}{1403}\right)\frac{\sin(h)}{\cos(h)}},$$

$$b = a\frac{\sin(h)}{\cos(h)}.$$

Some of the complications in this step stem from the fact that the variable t might be 0 in the denominator of p_1. Likewise, the distinction of cases in $\sin(h)$ and $\cos(h)$ is necessary to avoid division by 0 in a and b.

It turns out that both of these problems can be avoided quite elegantly. Consider, in the case $|\sin(h)| \geq |\cos(h)|$:

$$p_1' := \frac{50000}{13}N_c N_{cb} e_t,$$

$$b = \frac{p_2(2 + p_3)\frac{460}{1403}}{\frac{p_1'}{t\sin(h)} + (2 + p_3)\frac{220}{1403}\frac{\cos(h)}{\sin(h)} - \frac{27}{1403} + p_3\frac{6300}{1403}}$$

$$= \frac{t\sin(h)p_2(2 + p_3)\frac{460}{1403}}{p_1' + t(2 + p_3)\frac{220}{1403}\cos(h) + t\sin(h)\frac{6588}{1403}}$$

$$= \frac{23t\sin(h)p_2}{23p_1' + 11t\cos(h) + 108t\sin(h)},$$

and

$$a = \frac{23t\cos(h)p_2}{23p_1' + 11t\cos(h) + 108t\sin(h)}.$$

Conveniently, the exact same expressions are retrieved in the case $|\cos(h)| > |\sin(h)|$. These expressions are always well-defined since

$$23p_1' + 11t\cos(h) + 108t\sin(h)$$

$$= \frac{23p_1'p_2}{R_a' + G_a' + \frac{21}{20}B_a' + 0.305} > 0.$$

In the algorithm, the value of t can be retrieved via α (1) from the input variables. Indeed, if the saturation correlate s is given, one has

$$\alpha := \left(\frac{s}{50}\right)^2\frac{A_w + 4}{c};$$

if M is given, one can compute $C := M/F_L^{0.25}$ and then

$$\alpha := \begin{cases} 0 & \text{if } J = 0, \\ \dfrac{C}{\sqrt{J/100}} & \text{otherwise.} \end{cases}$$

It is mildly unfortunate that one has to introduce a case distinction for $J = 0$ here, but this is an operation that can still be performed at reasonable efficiency.
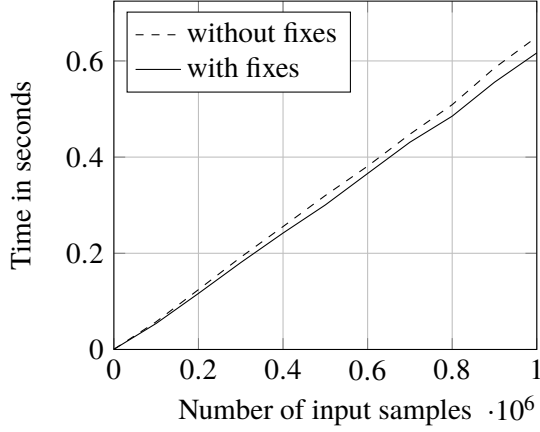
Figure 1: Performance comparison of the conversion from CAM16 to XYZ (with J, C, and h), implemented in colorio [4]. The suggested improvements in the inverse model lead to a speed-up of about 5%.

A. Full model

For the convenience of the reader, both forward and inverse steps of the improved CAM16 algorithm are given here. The wording is taken from [3] where applicable. The steps that differ from the original model are marked with an asterisk (*).

As an abbreviation, the bold letter $\boldsymbol{R}$ is used whenever the equation applies to R, G, and B alike.

Illuminants, viewing surrounds set up and background parameters (See the note at the end of Part 2 of Appendix B of [3] for determining all parameters.)

- Adopted white in test illuminant: X_w, Y_w, Z_w

- Background in test conditions: Y_b

- Reference white in reference illuminant:

$X_{wr} = Y_{wr} = Z_{wr} = 100$, fixed in the model

- Luminance of test adapting field (cd m^{-2}): L_A. L_A is computed using

$$L_A = \frac{E_W}{\pi} \frac{Y_b}{Y_W} = \frac{L_W Y_b}{Y_W},$$

where $E_W = \pi L_W$ is the illuminance of reference white in lx; L_W is the luminance of reference white in cd m^{-2}; Y_b is the luminance factor of the background; and Y_w is the luminance factor of the reference white.

- Surround parameters are given in Table 2: To determine the surround conditions see the note at the end of Part 1 of Appendix A of [3].

- N_c and F are modelled as a function of c, and their values can be linearly interpolated, using the data from 2.

Let M_{16} be given by

$$M_{16} := \begin{pmatrix} 0.401288 & 0.650173 & -0.051461 \\ -0.250268 & 1.204414 & 0.045854 \\ -0.002079 & 0.048952 & 0.953127 \end{pmatrix}.$$

A.1. Forward model

Step 0*. *Calculate all values/parameters which are independent of the input sample.*

$$\begin{pmatrix} R_w \\ G_w \\ B_w \end{pmatrix} = M_{16} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix},$$

$$D = F \left[1 - \tfrac{1}{3.6} \exp\left(\tfrac{-L_a - 42}{92} \right) \right].$$

4

	F	c	N_c
Average	1.0	0.69	1.0
Dim	0.9	0.59	0.9
Dark	0.8	0.525	0.8

Table 2: Surround parameters.

	Red	Yel-low	Green	Blue	Red
i	1	2	3	4	5
h_i	20.14	90.00	164.25	237.53	380.14
e_i	0.8	0.7	1.0	1.2	0.8
H_i	0.0	100.0	200.0	300.0	400.0

Table 3: Unique hue data for calculation of hue quadrature.

If D is greater than one or less than zero, set it to one or zero, respectively.

$$D_{\boldsymbol{R}} = D\frac{Y_W}{\boldsymbol{R}_W} - 1 + D,$$

$$k = \frac{1}{5L_A + 1},$$

$$F_L = k^4 L_A + 0.1(1 - k^4)^2 (5L_A)^{1/3},$$

$$n = \frac{Y_b}{Y_W},$$

$$z = 1.58 + \sqrt{n},$$

$$N_{bb} = \frac{0.725}{n^{0.2}},$$

$$N_{cb} = N_{bb},$$

$$\boldsymbol{R}_{wc} = D_{\boldsymbol{R}}\boldsymbol{R}_w,$$

$$\boldsymbol{R}_{aw} = 400\frac{\left(\frac{F_L \boldsymbol{R}_{wc}}{100}\right)^{0.42}}{\left(\frac{F_L \boldsymbol{R}_{wc}}{100}\right)^{0.42} + 27.13},$$

$$A_w = \left(2R_{aw} + G_{aw} + \frac{1}{20}B_{aw}\right) \cdot N_{bb}.$$

Step 1. *Calculate 'cone' responses.*

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = M_{16} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Step 2. *Complete the color adaptation of the illuminant in the corresponding cone response space (considering various luminance levels and surround conditions included in D, and hence in D_R, D_G, and D_B).*

$$\boldsymbol{R}_c = D_{\boldsymbol{R}} \cdot \boldsymbol{R}$$

Step 3*. *Calculate the modified postadaptation cone response (resulting in dynamic range compression).*

$$\boldsymbol{R}'_a = 400\,\mathrm{sign}(\boldsymbol{R}_c)\frac{\left(\frac{F_L|\boldsymbol{R}_c|}{100}\right)^{0.42}}{\left(\frac{F_L|\boldsymbol{R}_c|}{100}\right)^{0.42} + 27.13}.$$

Step 4. *Calculate Redness–Greenness (a), Yellowness–Blueness (b) components, hue angle (h), and auxiliary variables.*

$$\begin{pmatrix} p'_2 \\ a \\ b \\ u \end{pmatrix} := \begin{pmatrix} 2 & 1 & \frac{1}{20} \\ 1 & -\frac{12}{11} & \frac{1}{11} \\ \frac{1}{9} & \frac{1}{9} & -\frac{2}{9} \\ 1 & 1 & \frac{21}{20} \end{pmatrix} \begin{pmatrix} R'_a \\ G'_a \\ B'_a \end{pmatrix},$$

$$h := \arctan(b/a).$$

(Make sure that h is between $0°$ and $360°$.)

Step 5. *Calculate eccentricity [e_t, hue quadrature composition (H) and hue composition (H_c)].*

Using the following unique hue data in table 3, set $h' = h + 360°$ if $h < h_1$, otherwise $h' = h$. Choose a proper $i \in \{1, 2, 3, 4\}$ so that $h_i \le h' < h_{i+1}$. Calculate

$$e_t = \tfrac{1}{4}\left[\cos(h'\pi/180° + 2) + 3.8\right]$$

which is close to, but not exactly the same as, the eccentricity factor given in table 3.

Hue quadrature is computed using the formula

$$H = H_i + \frac{100e_{i+1}(h' - h_i)}{e_{i+1}(h' - h_i) + e_i(h_{i+1} - h')}$$

and hue composition H_c is computed according to H. If $i = 3$ and $H = 241.2116$ for example, then H is between H_3 and H_4 (see table 3 above). Compute $P_L = H_4 - H = 58.7884$; $P_R = HH_3 = 41.2116$ and round P_L and P_R values to integers 59 and 41. Thus, according to table 3, this sample is considered as having 59% of green and 41% of blue, which is the Hc and can be reported as 59G41B or 41B59G.

Step 6*. *Calculate the achromatic response*

$$A := p_2' \cdot N_{bb}.$$

Step 7. *Calculate the correlate of lightness*

$$J := 100(A/A_w)^{cz}.$$

Step 8. *Calculate the correlate of brightness*

$$Q := \frac{4}{c}\sqrt{\frac{J}{100}}(A_w + 4)F_L^{0.25}.$$

Step 9*. *Calculate the correlates of chroma (C), colorfulness (M), and saturation (s).*

$$t := \frac{50000/13N_cN_{cb}e_t\sqrt{a^2 + b^2}}{u + 0.305},$$

$$\alpha := t^{0.9} \cdot (1.64 - 0.29^n)^{0.73},$$

$$C := \alpha\sqrt{\frac{J}{100}},$$

$$M := C \cdot F_L^{0.25},$$

$$s := 50\sqrt{\frac{\alpha c}{A_w + 4}}.$$

A.2. Inverse model

Step 1. *Obtain J, t, and h from H, Q, C, M, s.*
    The input data can be different combinations of perceived correlates, that is, J or Q; C, M, or s; and H or h. Hence, the following sub-steps are needed to convert the input parameters to the parameters J, t, and h.

Step 1–1. *Compute J from Q (if input is Q)*

$$J := 6.25\frac{cQ}{(A_w + 4)F_L^{0.25}}.$$

Step 1–2*. *Calculate t from C, M, or s.*

- *If input is C or M:*

$$C := M/F_L^{0.25} \text{ if input is } M$$

$$\alpha := \begin{cases} 0 & \text{if } J = 0, \\ \dfrac{C}{\sqrt{J/100}} & \text{otherwise.} \end{cases}$$

- *If input is s:*

$$\alpha := \left(\frac{s}{50}\right)^2\frac{A_w + 4}{c}$$

Compute t from α:

$$t := \left(\frac{\alpha}{(1.64 - 0.29^n)^{0.73}}\right)^{1/0.9}$$

Step 1–3. *Calculate h from H (if input is H). The correlate of hue (h) can be computed by using data in table 3 in the forward model. Choose a proper $i \in \{1, 2, 3, 4\}$ such that $H_i \leq H < H_{i+1}$. Then*

$$h' = \frac{(H - H_i)(e_{i+1}h_i - e_ih_{i+1}) - 100h_ie_{i+1}}{(H - H_i)(e_{i+1} - e_i) - 100e_{i+1}}.$$

Set $h = h' - 360°$ if $h' > 360°$, and $h = h'$ otherwise.

Step 2*. *Calculate e_t, A, p_1', and p_2'*

$$e_t = \tfrac{1}{4}(\cos(h\pi/180° + 2) + 3.8),$$

$$A = A_w(J/100)^{1/(cz)},$$

$$p_1' = e_t\frac{50000}{13}N_cN_{cb},$$

$$p_2' = A/N_{bb}.$$

Step 3*. *Calculate a and b*

$$\gamma := \frac{23(p_2' + 0.305)t}{23p_1' + 11t\cos(h) + 108t\sin(h)},$$

$$a := \gamma\cos(h),$$

$$b := \gamma\sin(h).$$

Step 4. *Calculate R'_a, G'_a, and B'_a.*

$$\begin{pmatrix} R'_a \\ G'_a \\ B'_a \end{pmatrix} = \frac{1}{1403} \begin{pmatrix} 460 & 451 & 288 \\ 460 & -891 & -261 \\ 460 & -220 & -6300 \end{pmatrix} \begin{pmatrix} p'_2 \\ a \\ b \end{pmatrix}.$$

Step 5*. *Calculate R_c, G_c, and B_c,*

$$\boldsymbol{R}_c = \mathrm{sign}(\boldsymbol{R}'_a)\frac{100}{F_L}\left(\frac{27.13|\boldsymbol{R}'_a|}{400-|\boldsymbol{R}'_a|}\right)^{1/0.42}.$$

Step 6. *Calculate R, G, and B from R_c, G_c, and B_c.*

$$\boldsymbol{R} = \boldsymbol{R}_c/D_{\boldsymbol{R}}.$$

Step 7. *Calculate X, Y, and Z (for the coefficients of the inverse matrix, see the note at the end of the appendix B).*

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M_{16}^{-1} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

References

[1] Nathan Moroney et al. "The CIECAM02 Color Appearance Model". In: *The Tenth Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, CIC 2002, Scottsdale, Arizona, USA, November 12-15, 2002*. 2002, pp. 23–27. URL: `http://www.ingentaconnect.com/content/ist/cic/2002/00002002/00000001/art00006`.

[2] Ming Ronnier Luo and Changjun Li. "CIECAM02 and Its Recent Developments". In: *Advanced Color Image Processing and Analysis*. Ed. by Christine Fernandez-Maloigne. New York, NY: Springer New York, May 2012, pp. 19–58. ISBN: 9781441961891, 9781441961907. DOI: `10.1007/978-1-4419-6190-7_2`.

[3] Changjun Li et al. "Comprehensive color solutions: CAM16, CAT16, and CAM16-UCS". In: *Color Res. Appl.* 42.6 (June 2017), pp. 703–718. ISSN: 0361-2317. DOI: `10.1002/col.22131`.

[4] Nico Schlömer. *nschloe/colorio v0.1.0*. Feb. 2018. DOI: `10.5281/zenodo.1172996`. URL: `https://github.com/nschloe/colorio`.