

Modifications of the Robertson method for calculating correlated color temperature to improve accuracy and speed

Doug Baxter*^a

Michael Royer*^{a,b}

Kevin Smet*^c

*Co-first authors. All authors contributed equally to this work.

^a Pacific Northwest National Laboratory, Portland, OR, USA

^b School of Civil and Construction Engineering, Oregon State University, Corvallis, OR, USA

^c Light and Lighting Laboratory, KU Leuven. Ghent, Belgium

Corresponding Author: Michael P. Royer, Pacific Northwest National Laboratory, 620 SW, 5th Ave, Suite 810, Portland, OR 97204, USA. E-mail: michael.royer@pnnl.gov

This is an archival copy of an article first published online March 2, 2023, in *LEUKOS*. Please cite as:

Baxter D, Royer M, and Smet K. Modifications of the Robertson Method for Calculating Correlated Color Temperature to Improve Accuracy and Speed. *LEUKOS*, 20(1), 55-66. DOI: 10.1080/15502724.2023.2166060

Abstract

Correlated color temperature (CCT) is one of the principal metrics used in the engineering and specification of lighting. One of the most common methods used for calculating CCT was first proposed by Robertson in 1968. It utilizes a 31-row lookup table (LUT) based on isotherm lines and a formula to interpolate CCT values between the lines. The original Robertson method is known to have modest errors in determining CCT, so many alternative methods demonstrating improved accuracy have been subsequently proposed. Rather than an entirely new method, this analysis proposes three changes to the original Robertson method: recalculation of the LUT with an increase in the resolution to an increment of 1%, a correction to address CCT errors in the region between isotherms with slopes of opposite sign, and the application of a Fibonacci or binary search algorithm to efficiently search the expanded LUT. These changes result in a CCT calculation method, denoted Robertson2022, that produces CCT errors less than 0.1 K throughout the range of 1,500 K to 40,000 K with Duv between -0.05 and 0.05.

Keywords: Correlated color temperature; Robertson; Fibonacci, lookup table

1. Introduction

Correlated color temperature (CCT) is well-defined (CIE 2011; IES 2020) and an integral part of the engineering and specification of lighting, but there is no exact method for determining the CCT of a light source given its spectral power distribution (from which tristimulus values and chromaticity coordinates are calculated). Consequently, many methods seeking ever-more-accurate estimation of CCT have been proposed (Mori et al. 1964; Robertson 1968; Schanda et al. 1978; Krystek 1985; Xingzhong 1987; McCamy 1992; Walter 1992; Hernández-Andrés et al. 1999; Gardner 2000; Ohno 2014; Li et al. 2016; Zhang 2019). An important and widely used subset of these methods (Robertson 1968; Ohno 2014; Li et al. 2016; Zhang 2019) use a lookup table (LUT) as an initial step in the calculation process, following the introduction of this technique by Robertson in 1968.

At a minimum, a LUT for CCT calculations includes a predefined sorted array of color temperature values that are accompanied by chromaticity coordinates for the corresponding Planckian radiator. Advanced methods may include more information, such as precomputed derivatives. The CCT calculation process begins by searching the LUT to find the entries for which the distance between the tabulated chromaticity coordinates and those of the test light sources are the smallest and second smallest (i.e., the CCT of the test sources is between two isothermperature lines). This works because distance as a function of color temperature is a unimodal function with a single minimum—as long as the distance from the Planckian locus (Duv) is reasonably limited to not greater than 0.05 units. The 1968 Robertson method uses a 31-row LUT based on reciprocal megakelvin (MK^{-1}) indices with an increment of 10 between 0 MK^{-1} and 100 MK^{-1} , and 25 between 100 MK^{-1} and 600 MK^{-1} . This covers the range from 1667 K to 100,000 K.

After the initial result from searching the LUT is determined, an interpolation formula is applied to provide a more accurate estimate of CCT (symbol T_c). The Robertson method uses the formula shown in Equation 1, where d is the distance between the test source and lookup table coordinate, T is the LUT color temperature value, and j is the index of the LUT.

$$T_c = \left[\frac{1}{T_j} + \frac{d_j}{d_j - d_{j+1}} \left(\frac{1}{T_{j+1}} - \frac{1}{T_j} \right) \right]^{-1} \quad (1)$$

This equation is a technically an approximation that uses linear distances instead of arc distances; because the distances being evaluated are very small the simplification is effective.

At the time, the combination of Robertson’s 31-row LUT and the interpolation formula was reported to provide “sufficient accuracy” for determining CCT (Robertson 1968; Wyszecki and Stiles 1982). Recent work has reported errors for the original Robertson method of “several kelvins” (Ohno 2014), 2.4 K (Prytkov and Kolyadin 2021), and 12.4 K (Zheleznikova 2020), with the variation due to the range and resolution of chromaticity coordinates being evaluated. This level of error is tolerable for most specification applications but is generally greater than what is desirable for scientific endeavors. Error calculations will subsequently be discussed further.

The International Commission on Illumination (CIE) does not have an explicit standard method for calculating CCT, but in CIE 17:2020 the Robertson method is provided as an example method for calculating CCT (CIE 2011). Standard NR X08-017 from Association Française de Normalisation (AFNOR),

Evaluation of the correlated color temperature of light sources, utilizes the Robertson method with additional specificity and updates for modern implementation (AFNOR 2017).

In 2014, Ohno proposed a new method for calculating CCT that uses the same concept as Robertson's method: a LUT with interpolation formulae. Ohno's LUT has 1% CCT increments, which is a substantial increase over Robertson. He also described additional accuracy gained from using 0.25% increments. Ohno replaced Robertson's interpolation formula with a combination of triangular and parabolic approaches, reducing the reported error to "mostly within 0.5 K" for the range of 2000 K to 20,000 K with Duv between -0.03 and 0.03, with a maximum error of 0.8 K. With the 0.25% resolution LUT, the error was reported as less than 0.2 K. Ohno's method has been used by both the Illuminating Engineering Society (IES) and CIE to compute CCT for color rendition metrics (CIE 2017; IES 2020).

More recently Li et al. (2016) demonstrated that doubling the resolution of Robertson's LUT increased the accuracy by about a factor of two—the absolute maximum errors they reported, for CCTs up to 1,000,000 K, are not particularly relevant for architectural lighting, especially given that CCT errors generally increase with CCT for the Robertson method. Likewise, Zheleznikova, Prytkov, and Kokinov (2020) reported that the Robertson method with new LUTs having increments of 5 MK⁻¹, 2.5 MK⁻¹, and 1 MK⁻¹, resulted in maximum errors of 0.6 K, 0.16 K, and 0.025 K for the range of 1667 K to 10,000 K with Duv between -0.05 and 0.05.

This work further explores the solution of reducing the LUT increment (i.e., increasing the number of rows in the LUT) to improve the accuracy of the original Robertson method. First, it introduces a correction for the original Robertson method to address anomalous CCT errors in the region between isotherms with slopes of opposite sign. It also documents the use of a Fibonacci search algorithm to more efficiently search the LUT—guidance on the search technique has increased importance with larger LUTs.

While the accuracy of existing CCT calculation methods is sufficient for specification circumstances—especially given the limited ability of CCT to describe perceived color of a light—increasing the accuracy and consistency of calculations does have value in some scientific and industrial cases, such as in computing color rendition metrics or determining "bins" for LEDs. As shown in this article, accuracy gains can be made while simultaneously improving the efficiency of the calculation.

2. Slope Sign Correction

During this investigation, it became apparent that a few test conditions near 1635 K—with the exact value depending on the LUT increment—resulted in unexpectedly large CCT errors. These were revealed by introducing higher resolution LUTs, such that the error did not end up as noise with the less accurate original LUT, and an expanded LUT range with a lower limit less than the original 1667 K. Previous work may have missed these errors by limiting the range of tested conditions.

It was discovered that these outlier errors occurred at inflection points where the sign of the slope of the isotherm lines changed. Specifically, for adjacent isotherm lines that have the same signed slopes, when the distance (d_j) changes sign at d_{j+1} it means the test chromaticity lies between isotherm lines j and $j + 1$. However, when the test chromaticity lies between the two isotherm lines on either side of the apex, denoted here as line a and line $a + 1$, it needs to be above both and $d_a \times d_{a+1}$ needs to be positive.

When searching the LUT, the isotherm line j that lies closest to the test chromaticity is found. If the slope (m_j) is positive, the coordinate of the closest Planckian is to the right of the apex and if d_j is positive, the bound is isotherm line $j + 1$; otherwise, it is line $j - 1$. If m_j is negative and d_j is positive, the bound is isotherm line $j - 1$; otherwise, it is line $j + 1$. So, if $d_j \times m_j$ is positive, the bounding lines are j and $j + 1$; otherwise, the bounding lines are j and $j - 1$.

If $j = a$ and $j + 1 = a + 1$ then the test coordinate is in between isotherms with slopes of opposite sign and the ratio $\frac{d_j}{d_j - d_{j+1}}$, which is part of Equation 1, needs to be replaced by $\frac{d_j}{d_j + d_{j+1}}$. This can be found when $m_j \times m_{j+1}$ is negative.

Note that several quantities used in the computation, including $\sqrt{1 + m_j^2}$, $\frac{1}{T_j}$, $\frac{1}{T_{j+1}} - \frac{1}{T_j}$, and $m_j \times m_{j+1}$ can be precomputed for each row of the LUT, saving computation time. This approach also reduces the need to sometimes check on both sides of an isoline for the second closest point, so this method can provide a very small overall time savings compared to an equivalent approach without the correction.

Table 1 provides errors with and without the correction using a 1% resolution LUT. Without the correction, the error for some coordinates is up to 10 K, but with the correction the error is less than 0.01 K, in line with the error for other points with similar chromaticity. While the large errors occur for only a small fraction of chromaticity coordinates (i.e., points outside this region have identical errors with or without the correction), they can inflate the maximum error found over a large set of test chromaticity coordinates. All subsequently reported analyses employ the correction.

An alternative to the slope sign correction method is to modify Equation 1 by taking the absolute value of each of the three d terms, as shown in Equation 2. This method produces equivalent results, but timing tests showed it was very slightly slower than the proposed method. Nonetheless, it could be a simpler solution in some circumstances. Speed comparisons between different CCT calculation methods are addressed in detail in a separate article.

$$T_c = \left[\frac{1}{T_j} + \frac{|d_j|}{|d_j| - |d_{j+1}|} \left(\frac{1}{T_{j+1}} - \frac{1}{T_j} \right) \right]^{-1} \quad (2)$$

3. LUT Resolution

As previously noted, the original LUT proposed by Robertson in 1968 utilizes only 31 rows, reflecting the computing ability at the time. Each row comprises a CCT value (between 1667 K to 100,000 K) and associated chromaticity coordinates. Today, a LUT with hundreds of rows can be searched in nanoseconds. Even decades ago, Wyszecki recognized that the “success of the method depends on choosing a set of isotherm lines with sufficiently narrow spacing so that the errors introduced...are negligible.” That is, the key to improving accuracy is increasing the resolution of the LUT, not necessarily changing the interpolation formula. This has been confirmed by Li et al. (2016) and Zheleznikova, Prytkov, and Kokinov (2020).

To further investigate the tradeoff between accuracy and LUT resolution, we examined 26 different resolution LUTs used with the Robertson method: original 10 MK⁻¹ and 25 MK⁻¹ combination, 25 MK⁻¹, 5 MK⁻¹, 2.5 MK⁻¹, 1 MK⁻¹, 1000 K, 500 K, 250 K, 100 K, 50 K, 25 K, 10 K, 1 K, 0.2 K, 0.1 K, 15%, 2%, 1.75%,

Table 1. CCT error with and without the identified correction in the region where the slope of the isoline changes sign. CCT calculated with a 1% increment LUT and all other elements the same.

Reference CCT (K)	Reference u	Reference v	CCT Error (K) Without Correction	CCT Error (K) With Correction
<i>All lower reference CCTs have equal error with and without correction</i>				
1609.63	0.34385	0.34329	0.002480747	0.002480747
1609.97	0.34378	0.33184	0.001570544	0.001570544
1615.69	0.34311	0.32756	0.012933420	0.001706517
1617.32	0.34296	0.34709	0.021284270	0.004161127
1619.06	0.34273	0.32457	0.006994951	0.002092184
1619.09	0.34273	0.32317	0.006330294	0.001926237
1622.80	0.34233	0.35389	0.065948130	0.005025245
1623.11	0.34227	0.31247	0.001530697	0.000525103
1625.59	0.34200	0.33044	0.024794790	0.001611575
1625.93	0.34196	0.35298	0.050822140	0.002786862
1626.59	0.34189	0.35216	0.047185230	0.002085962
1627.72	0.34176	0.34848	0.039159150	0.000741710
1631.19	0.34139	0.31751	4.944773000	0.000564678
1632.47	0.34126	0.31232	6.852653000	0.000222083
1633.58	0.34110	0.33946	8.408778000	0.003258194
1635.00	0.34089	0.36447	10.300570000	0.006405025
1643.00	0.34006	0.33326	0.001061939	0.001061939
1645.36	0.33984	0.32250	0.000228114	0.000228114
<i>All higher reference CCTs have equal error with and without correction</i>				

1.5%, 1.25%, 1%, 0.75%, 0.5%, 0.25%. Note that the reciprocal megakelvin LUTs had a range from 10 MK-1 to 1025 MK-1, except for the original LUT which starts from 10 MK-1 and runs to 625 MK-1 and the extended version of that original LUT which also runs to 1025 MK-1. The other LUTs began at 1000 K and extended to 40,000 K. A primary goal was to identify the LUT with the smallest number of rows that could result in a maximum error of less than 0.1 K. The 0.1 K criterion was determined through a consensus process of the IES Color Committee and is addressed further in the discussion section.

The extents and increment of the calculation, as well as the constant used to calculate Planckian radiation can change the calculated CCT (Borbély et al. 2001). Each LUT was created by computing the coordinates of Planckian radiation using spectral data from 360 nm to 830 nm in increments of 1 nm. For computer Planckian radiation, c_2 was $1.4388 \cdot 10^{-2}$ m K and the refractive index n was 1.0.

A set of 1,270 reference chromaticity coordinates ranging between CCTs of 1500 K and 40,000 K and Duv values of -0.05 to 0.05 (excluding any beyond the spectrum locus) were computed using predetermined CCT and Duv values, following the derivative method described by Mori et al. (1964) and Robertson (1968), as opposed to the less accurate “offset” method used by Ohno (2014) or numerical central derivative (finite differences) method of Prytkov and Kolyadin (2021). The specific points were derived from an approximately even grid in chromaticity space, with nominal CCT increments of 50 K between

1500 K and 2000 K, 100 K between 2000 K and 2800 K, 3000 K, 250 K increments between 3250 K and 5000 K, 5500 K, 6000 K, 6500 K, 7000 K, 8000 K, 9000 K, 10000 K, 12500 K, 15000 K, 20000 K, 30000 K, and 40000 K. At each nominal CCT there were five nominal Duvs, with 0.025 units increments covering the range. This arrangement provides full coverage of the relevant area and the extents of the LUT. Importantly, a random component was added to each nominal target coordinate (CCT_i, Duv_i), to avoid biasing the results by generating ten random (CCT, Duv) value pairs within the following intervals: $[(X_{i-1} - X_i)/2, (X_{i+1} - X_i)/2]$, with X representing either CCT or Duv . The intervals were sampled using a uniform distribution. The reference coordinates are available in a supplemental file.

Error was determined by comparing the computed CCT from the result set of 1,270 chromaticity coordinates using a given LUT. As shown in Fig. 1 and Table 2—which are based on calculations performed in Python—the Robertson method with a 1% increment LUT (and the slope sign correction) was the lowest resolution that resulted, within the test range, in a maximum error of less than 0.1 K. This combination is subsequently denoted Robertson2022 in this article. The maximum error for the Robertson2022 method was 0.097 K; all error data is available in a supplemental file. With a range of 1000 K to 40,107 K (the next number over 40,000 K), the LUT for Robertson2022 has 372 rows, and in .csv format is 70 KB. As shown in Fig. 1 and Table 2, increasing the LUT resolution can improve the accuracy further, but the practical return was considered minimal as CCT is rarely reported with a

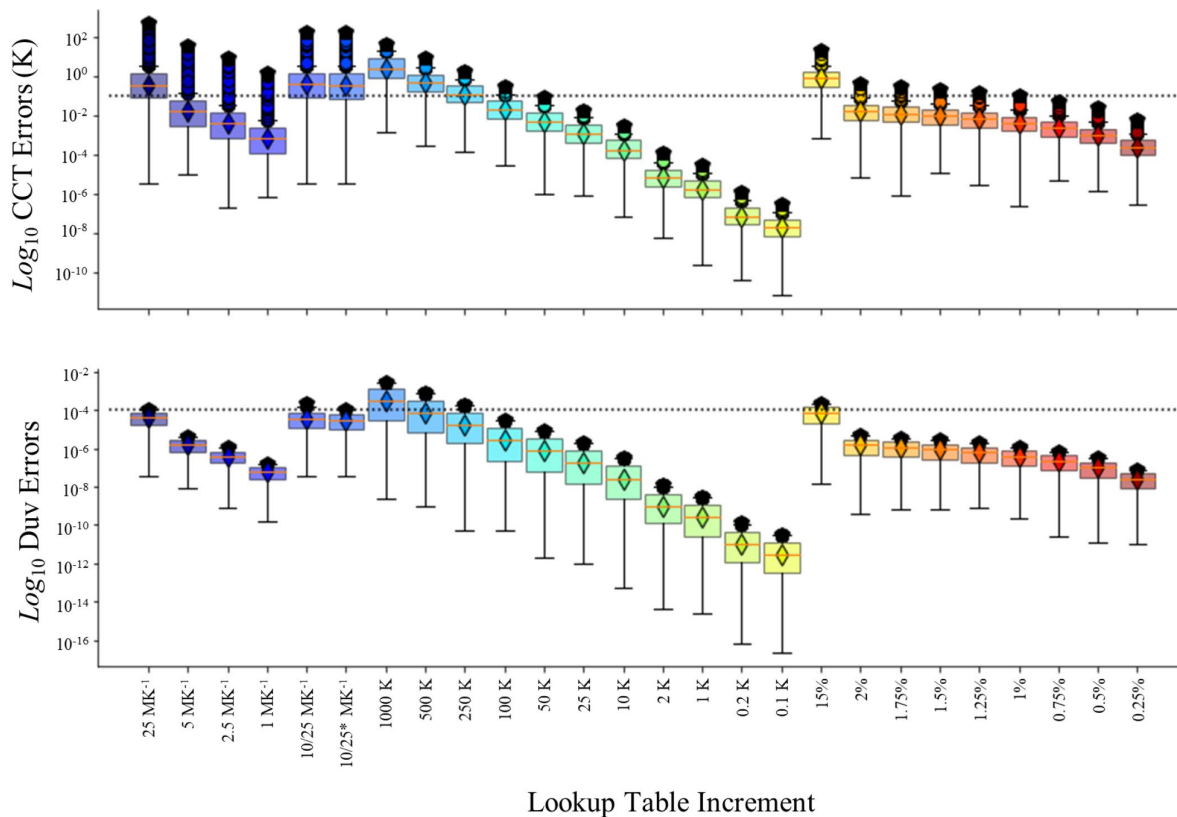


Fig. 1. Boxplots of the CCT and Duv errors for the Robertson method (with slope correction) using different LUTs. Increment 10/25 denotes the original LUT and 10/25* denotes the extended original LUT. The error thresholds for CCT (0.1 K) and Duv (0.0001) are indicated by the dotted line in each chart.

Table 2. Maximum errors using the documented reference set for the Robertson method (with slope correction) using different LUTs covering the range of 1000 to 40,000 K, except for the original 1968 method.

LUT Increment	LUT Rows	Maximum CCT Error (K) ¹
10 & 25 MK ⁻¹ (Original 1968)	31	151.7
10 & 25 MK ⁻¹ (Original extended)	47	151.7
25 MK ⁻¹	42	437.5
5 MK ⁻¹	202	33.1
2.5 MK ⁻¹	402	8.4
1 MK ⁻¹	1,002	1.5
1000 K	40	39.1
500 K	79	7.6
250 K	157	1.80
100 K	391	0.31
50 K	781	0.076
25 K	1,561	0.019
10 K	3,901	0.0030
2 K	19,501	0.00013
1 K	39,001	0.000029
0.2 K	195,001	0.0000012
0.1 K	390,001	0.00000029
15%	28	18.8
2%	188	0.40
1.75%	214	0.29
1.5%	249	0.20
1.25%	298	0.13
1%	372	0.097
0.75%	495	0.050
0.50%	741	0.025
0.25%	1,479	0.0061

¹ Maximum errors were always rounded up.

precision greater than whole numbers. Specific applications may utilize a LUT that is appropriate for the level of accuracy that is needed, either higher or lower.

Although not the focus of this work, Duv errors were also calculated, and are shown in Fig. 1. All methods that provided CCT error less than 0.01 K provided Duv error less than 0.0001.

For comparison, Ohno's method with a 1% increment LUT had maximum errors of 2.7 K over the same range, and the 0.25% increment LUT had a maximum error of 0.4 K. That is, a higher resolution LUT is needed for the Ohno method to achieve the same accuracy as Robertson. Thus, Robertson (with slope correction) has the more accurate interpolation method.

If increased accuracy is needed, a higher resolution LUT (e.g., 1 K increments) can be used with the Robertson method. It is also possible to use iterative methods, such as those proposed by Li et al. (2016) or Zhang (2019). Li et al. documented the application of a Newton Raphson method and Zhang proposed the use of "cascading" LUTs, with successively refined LUT determined to zero in on the CCT of interest.

Both methods have increased calculation complexity. Calculation speed comparisons between all potential methods is not within the scope of this article but has been completed for a separate article.

4. LUT Search Method

There are several ways to search a LUT that will all return the correct answer, but the number of calculations needed to reach the solution and type of operations being performed vary, which can affect the time needed to complete the calculation. Important search methods that may be used include:

1. Linear: A linear search means the distance between the chromaticity of the test light source and the chromaticity coordinates in the LUT is calculated for each row of the LUT in sequential order until the point where the distance begins to increase (one row after the minimum). A linear search is simple to program and is used, for example, in Excel tools supplied with ANSI/IES TM-30 (IES 2020) and CIE 224 (CIE 2017)
2. Binary (or bisection search): A binary search progresses by repeatedly dividing the array in half until the minimum distance is found. The first, last, and middle distance are first calculated, which allows identification of the portion of the array in which the minimum falls, producing the next interval to be searched. Executing a binary search provides an overall time savings compared to a linear search. Some implementations of CCT calculations, such as those in LuxPy (Smet 2020), implement a binary search.
3. Golden section: A golden section search algorithm also divides the array into parts, but unlike the binary method, they are unequal. The method maintains four points, with the three intervals having the golden ratio. As with a binary search, the minimum is found by iteratively reducing the region being searched. Zhang (2019) documented the use of a golden section search, in combination with a technique of iteratively creating new LUTs within a specified region, to estimate CCT at an accuracy determined by a stopping criterion.
4. Fibonacci: The Fibonacci search algorithm is closely related to the golden section search. The relative size of the two parts of the searched array is the same as consecutive Fibonacci numbers—this ratio approaches the golden ratio. The size of the subintervals is not perfectly constant, however.

Other search algorithms also exist, but a complete description of all methods is not within the scope of this article.

In many practical uses of CCT, the calculation time—regardless of search method or algorithm—is not an important factor because it is already minimal. Perhaps because of this, most of the proposed CCT calculation methods that utilize a LUT have not explicitly recommended a search method. Nonetheless, in some research applications minimizing CCT calculation time is critical and all efforts to save even a small fraction of a second are worthwhile.

Except for linear search, the methods mentioned above examine selected entries in the LUT and based on the distance between the test coordinates and those entries determine a reduced range of rows that will contain the closest point in the table to the test chromaticity. They all take advantage of the fact that the distance from a target point to a point on the Planckian locus (representing row in the LUT) is a unimodal function. This section subsequently compares the binary and Fibonacci search methods, providing specific documentation for how to use the Fibonacci search in this application.

Let γ be the number of *distance comparisons* needed by a method for a LUT with n rows. Then for a binary search, γ is given by Equation 2, where *ceil* denotes the ceiling function (smallest integer greater than), and \log_2 is the base 2 logarithm.

$$\gamma = \text{ceil}(2\log_2 n) \quad (2)$$

For the Fibonacci search technique—which is described further below— γ is given by Equation 3.

$$\gamma = \text{ceil}(1.45\log_2 n) + 1 \quad (3)$$

Thus for n greater than eight rows, a Fibonacci search will need fewer distance calculations than a binary search. Additionally, the binary search needs to use a shift and an addition to determine the row index for each distance evaluation, whereas the Fibonacci search only needs an addition. (The golden section search is very similar to a Fibonacci search, but the index calculation requires an addition and a multiplication and then a truncation for its row index calculation.) Note that the number of distance calculations is only partially explained by the number of intervals searched. In many cases, a binary search may lead to fewer intervals being evaluated but requires more distance calculations.

To compare the performance of the binary and Fibonacci search methods for CCT calculations, both were implemented in code written in C and executed on a computer with a 2.8 GHz AMD EPYC 7282 processor. The same set of 1,270 reference chromaticity coordinates previously described was used. Twenty trials of the Robertson2022 doing two billion CCT and Duv calculations and the 20 more doing one billion calculations using either the binary search or Fibonacci search were timed. For each pair a time difference between two billion and one million calculations was used to isolate just the time for the CCT and Duv calculation, and ultimately determine the mean time per single CCT calculation. During the trials, the reference set was continually cycled after a random starting point, although with either Fibonacci or binary searches the position of the test coordinate within the range of the LUT does not affect the number of distance calculations performed, and thus does not affect the calculation time.

For this test, the Fibonacci search method led to an approximately 24% reduction in overall CCT and Duv calculation time (not just the LUT search) compared to the binary search. One CCT calculation, excluding the table load, took 80 nanoseconds using the Fibonacci search (Equation 1) or 81.5 nanoseconds using Equation 2. It took 104 nanoseconds using the binary search. While the absolute time reduction for a single CCT calculation is not practically meaningful, for a scientific endeavor requiring millions of CCT calculations, a 24% reduction in calculation time is extremely valuable.

Each distance evaluation takes approximately 4.5 nanoseconds. To reduce the time further, the LUT would have to be reduced to less than 233 rows for a 4.5 nanosecond savings per (u, v) to (CCT, Duv) conversion. At the same resolution, that would require reducing the range to 1500 K to 10,000 K. Alternatively, the resolution could be increased to approximately 1.6% with the same range covered, which would increase the maximum error to greater than 0.2 K

4.1 Details of the Fibonacci search

In the application of CCT calculations, there is a LUT of n rows indexed from 0 to $n - 1$. The color temperature, u coordinate, and v coordinate can be denoted as t_k , u_k , and v_k for the k^{th} row, with k in $[0, n - 1]$. Note that this discussion focuses on the three primary columns; the algorithm can be

modified, as discussed later, to work on tables with differing numbers of columns. Speed-optimized implementations of the methods of Robertson, Ohno, and Li et al. can use different precomputed auxiliary data that is stored in the LUT.

Given chromaticity coordinates (u_x, v_x) , the goal is to find the index, k , for which the distance from (u_x, v_x) to (u_k, v_k) is a minimum. It suffices to find the minimum square of the distance between (u_x, v_x) and (u_k, v_k) as $a^2 < b^2$ implies $|a| < |b|$. This saves needing to take square roots for each distance computed.

For convenience, the squared distance can be defined as the function g as shown in Equation 4:

$$g(u_k, v_k, u_x, v_x) = (u_k - u_x)^2 + (v_k - v_x)^2 \quad (4)$$

An additional requirement for the Fibonacci search technique is an array of Fibonacci numbers, $F[]$, that omits the traditional first two elements of the sequence, 0 and 1, and starts instead with first two elements 1 and 2 as shown in Equations 5 through 7.

$$F[0] = 1 \quad (5)$$

$$F[1] = 2 \quad (6)$$

$$F[i] = F[i - 1] + F[i - 2], i \geq 2 \quad (7)$$

The length of F , $l + 1$, is chosen such that Equations 8 and 9 hold:

$$F[l - 1] < n - 1 \quad (8)$$

$$F[l] \geq n - 1 \quad (9)$$

Then during each stage of the search four indices are tracked into the table: A *left* and a *right* index that define the limits of the index range in which the point of minimum distance from the target lies and two indices a and b that lie between *left* and *right* and are successive Fibonacci numbers more than *left*. Each step of the search then involves reducing the distance between *left* and *right* to the next lower Fibonacci number by replacing one of the endpoints with the interior point at largest distance from the target and choosing a new interior point that maintains the relationships in Equations 10 to 12 with j decreasing at each stage.

$$a = \textit{left} + F[j] \quad (10)$$

$$b = \textit{left} + F[j + 1] \quad (11)$$

$$\textit{right} = \textit{left} + F[j + 2] \quad (12)$$

Note that when the row k closest to the target is greater than $F[l - 1]$ as defined in Equations 8 and 9 and $F[l]$ is strictly greater than $n - 1$, Equations 10 to 12 may generate indices $> n - 1$ and so additional restrictions must be placed on a , b , and *right* to avoid referring to nonexistent rows of the LUT, and to maintain strict inequality of a and b as defined in Equations 13 to 15:

$$\textit{right} \leq n - 1 \quad (13)$$

$$b \leq n - 1 \quad (14)$$

$$a < b \quad (15)$$

The *left* and *right* indices are initialized to 0 and $n - 1$ respectively, and then only set to successive a or b values, hence Equation 13 always holds if Equation 14 is enforced. As b is updated, Equation 14 must be enforced and as a is updated, Equation 15 must be enforced. Note that when j is 0, *left*, a , and b necessarily point to consecutive rows in the LUT and *right* is either one more than b or equal to b if the solution is at the very end of the table. Thus, the minimum distance row from the table must be either a or b if *left* is not 0 and *right* is not $n - 1$. The latter conditions are ensured by extending the LUT downward and upward so that (u_0, v_0) and (u_{n-1}, v_{n-1}) are never the closest points to a target (u_x, v_x) .

A pseudo-code algorithm for the Fibonacci search as described is given in . The variables *dsq_left*, *dsq_right*, *dsq_a*, and *dsq_b* are used to hold the distances from (u_x, v_x) to the points in the corresponding rows of the table so as to compute the distance to any given row in the table only once.

Note that only one of line 32 or line 20 is executed; thus, only one distance computation (call to *g*) is made per iteration of the loop. In practice, the LUT may have a different number of columns depending on the method of interpolation to be used, and it is useful to store it in a single linear array of length $m * n$, where m is the number of columns. Then it saves a little bit of indexing time to have the search generate $k * m$ instead of k , so as to get the offset of the first element of the k^{th} row of the table. This is easily accomplished by starting with the Fibonacci array generation replacing Equations 5 and 6 with Equations 16 and 17.

$$F[0] = m; \tag{16}$$

$$F[1] = m + m; \tag{17}$$

Then, Equation 7 can be used to generate the rest of the table until $F(l) \geq m * (n - 1)$. Also, line 18 in Fig. 2 needs to change to $a = b - m$, Line 29 needs to change to $b = right - m$, and the $(n - 1)$ in lines 28 and 29 of Fig. 2 needs to change to $m * (n - 1)$.

5. Discussion

The original method for estimating CCT proposed by Robertson in 1968 has been challenged by many researchers. This article demonstrates that with three small changes, the Robertson method can be highly accurate and computationally efficient.

First, this work identified a small correction to improve the interpolation algorithm in a specific region of the chromaticity diagram where the slope of the isotherms change sign. The need for this correction arose from testing lower CCT values than have typically been tested in the past in combination with the use of higher resolution LUTs, which increase the accuracy of the calculation to a point where the errors in the anomalous region become noticeable. For a 1% increment LUT, it reduced the maximum error found in the set of 1,270 reference coordinates from 10.3 K to 0.097 K. This correction enables a lower-resolution LUT to achieve the same level of maximum error.

As was previously reported by others, there is a direct relationship between the resolution of the LUT and the accuracy of the CCT estimation. This work offers an expanded look at the relationship. It

```

left = 0; 1
right = n - 1; 2
a = F[l - 2]; 3
b = F[l - 1]; 4
dsq_left = g(u0, v0, ux, vx); 5
dsq_a = g(ua, va, ux, vx); 6
dsq_b = g(ub, vb, ux, vx); 7
dsq_right = g(un-1, vn-1, ux, vx); 8
for (j = l - 3; j ≥ 0; j = j - 1) { 9
    if (dsq_a < dsq_b) { 10
        /* row a was closer shift to [left, b] */ 11
        right = b; 12
        dsq_right = b; 13
        b = a; 14
        dsq_b = dsq_a; 15
        a = left + F[j]; 16
        if (a > b) { 17
            a = b - 1; 18
        } 19
        dsq_a = g(ua, va, ux, vx) 20
    } else { 21
        /* row b was closer, shift to [a, right] */ 22
        left = a; 23
        dsq_left = dsq_a; 24
        a = b; 25
        dsq_a = dsq_b; 26
        b = left + F[j + 1]; 27
        if (b ≥ (right)) { 28
            b = (right - 1); 29
        } 30
        dsq_b = g(ub, vb, ux, vx); 31
    } 32
} 33
if (dsq_a < dsq_b) { 34
    k = a; 35
} else { 36
    k = b; 37
} 38
/* k is the index of the row of the table closest to (ux, vx) */ 39

```

Fig. 2. Pseudocode for a Fibonacci search of a CCT LUT.

corroborated earlier findings, but also found substantially higher maximum errors using the prescribed reference set (1500 K to 40,000 K and -0.05 to 0.05 Duv with random offsets from target coordinates). Zheleznikova, Prytkov, and Kokinov (2020) reported the maximum error for LUTs with increments of 5 MK⁻¹, 2.5 MK⁻¹ and 1 MK⁻¹ to be 0.6 K, 0.16 K, and 0.025 K respectively, over the range of 1667 K to 10,000 K with Duv between -0.05 and 0.05, whereas this analysis found maximum errors of 33.1 K, 8.4 K,

and 1.5 K for the same resolution LUTs. The key difference is that this analysis extended the range of CCT considered, and the method for establishing the reference set was also different.

This work evaluated LUTs based on reciprocal megakelvin increments, equal kelvin increments, and percentage increments. Both reciprocal megakelvin increments and percentage increments lead to variable kelvin differences between sequential LUT rows, in this case increasing with higher color temperatures. This can be an advantage, because CCT increments are not constant in chromaticity diagrams; the chromaticity difference becomes smaller as CCT increases. Likewise, reciprocal megakelvin units are closer to having equal perceptual increments compared to kelvin increments (Priest 1933; Ashdown 2002).

Using a percentage increment or reciprocal megakelvin increment typically leads to slightly higher CCT errors (expressed in kelvin) as CCT increases, but the increment can be tailored so the maximum error still meets any *a priori* target, such as 0.1 K. With this approach, the error at CCTs more commonly used in architectural lighting (e.g., 2400 K to 6500 K) will be considerably smaller than at CCTs exceeding 10,000 K. For the Robertson2022 method, the maximum error for a reference color temperature under 10,000 K was 0.024 K.

Table 2 illustrates that for a similar number of rows in the LUT, a percentage increment has substantially lower maximum error. For example, a 1% increment LUT with 372 rows had a maximum error of 0.097 K, but a 100 K increment LUT with 391 rows had a maximum error of 0.31 K. A 2.5 MK⁻¹ LUT with 402 rows had a maximum error of 8.4 K. Thus, using a percentage increment appears to be the method that results in the lowest maximum error for a given LUT size.

With the implementation of higher-resolution LUTs, the method used to search the LUT becomes a more important consideration. While any method can be used without a notable difference in speed for a single calculation, many research applications requires billions or more CCT calculations at a time. This work demonstrated that for LUTs greater than eight rows—always the case for CCT calculations—the Fibonacci search requires fewer distance calculations than a binary search, even if it requires slightly more index values. In an implementation in the programming language C with a 1% LUT, searching the LUT with a Fibonacci method was found to reduce the overall CCT calculation time by 10% compared to searching the LUT with a binary method.

Importantly, the finding regarding search methods may not hold for all different programs. For example, an implementation in Python can use Numpy, which includes a binary search but not a Fibonacci search. A binary search using Numpy (which uses C) is faster than a Fibonacci search done directly in Python. Fortunately, the choice of search method does not alter the result of the calculation, just the steps taken to get there, and thus the speed. Those implementing the method with a need to minimize calculation time should carefully consider the programming language used and choose the search method that is best for the specific case.

Addressing the perceptual meaningfulness of CCT or the limits of its relevance as a measure of white light are outside the scope of this article. However, it is worth noting that some researchers have essentially concluded that increasingly “accurate” CCT calculations are not needed because CCT lacks perceptual meaningfulness to begin with (Grum et al. 1978; Borbély et al. 2001). Borbély et al. recommended not reporting CCT values in any finer resolution than what has essentially become the nominal designation of ANSI C78.377 (NEMA 2017) or roughly 500 K. Modern practices, such as “1-step”

binning, already indicate this is too coarse, but there is a legitimate question regarding how high of accuracy is needed.

The alterations to the original 1968 Robertson method described in this article can be applied with any resolution LUT, meeting any desired level of CCT accuracy. We favor the use of a 1% LUT, delivering CCT accuracy of 0.1 K or greater, for its balance between speed and accuracy. It provides more than three orders of magnitude greater accuracy than the original 1968 Robertson method, and if carefully implemented, the Robertson2022 may not require any additional calculation time. Consider that a linear search applied to the original 31-row LUT would require, on average 15 distance calculations to find the minimum if the distribution of CCTs being calculated is even over the entire range. Due to the improved search method, the Robertson2022 method actually requires one fewer, at 14. As implemented in the pseudocode, the Robertson2022 search involves only computing d_j , eliminating the need to look for the sign change at each step of a linear search, or having to look for the sign change on either side of the minimum d_j after it is found. The improvement of knowing whether to choose isothermal line $j + 1$ or $j - 1$ based on the agreement or disagreement of the sign of d_j and m_j saves an extra bit of computation. With the additional data needed for improved accuracy precomputed—only requiring 70 KB—and not causing increased calculation time on average, Robertson2022 does not have a measurable or practical downside compared to the original 1968 Robertson method.

6. Conclusion

This work investigated improvements to the Robertson method for calculating correlated color temperature (CCT) of a light source, which was first introduced in 1968. CCT estimation accuracy was investigated using a set of 1,270 reference (u, v) coordinates with CCTs between 1500 K and 40,000 K and Duvs between -0.05 and 0.05.

Three improvements are proposed to increase the accuracy and improve the calculation efficiency:

1. Implement a numerical correction to address outlier CCT errors at the point where adjacent isotherm lines included in the LUT have different signs. For a LUT with a 1% increment, this correction reduced the maximum error for the set of reference coordinates from 10.3 K to 0.097 K. The correction can be achieved by precomputing values and storing them with the LUT.
2. Increase the resolution of the LUT from a combination of 10 MK⁻¹ and 25 MK⁻¹ (original Robertson 1968 method) to a resolution of 1% increments. With the slope sign correction, the maximum error found for the reference coordinates with a 1% increment LUT was 0.097 K, which is substantially less than the 151.7 K found for the original method. While higher accuracy is possible with higher-resolution LUTs, additional accuracy is likely unneeded. Greater benefit could come from lighting standards organizations adopting a single recommended method, which would allow all users who follow the guidance to perform the calculation in the same way—something that does not occur today.
3. In applications where calculation speed is a consideration, the method used for the initial search of the LUT to find the row with the chromaticity coordinates closest to those of the test source should be carefully considered. All else equal, this work identified that a Fibonacci search requires fewer calculations and thus is faster than a binary search. However, the fastest non-linear search method may vary for implementations in different software packages. On average, using a non-linear search of the expanded 1%-increment LUT will require less computations than a linear-search of the much lower resolution original 1968 LUT. That is, the proposed

Robertson2022 method offers orders of magnitude more accuracy estimates of CCT without adding calculation time.

Implementing these three improvements results in the Robertson method having one of the best accuracy-to-speed relationships of all previously proposed methods for estimating CCT, and it should be considered a strong candidate for standardization by lighting standards organizations, which currently do not endorse any particular method.

Acknowledgements

The authors gratefully acknowledge the input from members of the IES Color Committee regarding the improvement and standardization of methods for calculating CCT.

Funding

The work of Michael Royer and Doug Baxter was supported by the U.S. Department of Energy's Lighting R&D Program, part of the Building Technologies Office within the Office of Energy Efficiency and Renewable Energy (EERE).

Disclosure Statement

The authors report there are no competing interests to declare.

References

- Ashdown IE. 2002. Chromaticity and color temperature for achitectural lighting. In: Solid State Lighting II. Vol. 4776. SPIE. p. 51–60.
- [AFNOR] Association Française de Normalisation. 2017. XP X08-017: Evaluation of the correlated color temperature of light sources.
- Borbély Á, Sámson Á, Schanda J. 2001. The concept of correlated colour temperature revisited. *Color Res Appl.* 26(6):450–457. doi:10.1002/col.1065.
- [CIE] Commission Internationale de l'Eclairage. 2011. International Lighting Vocabulary. CIE S017/E:2020. Vienna, Austria: Commission Internationale de l'Eclairage.
- [CIE] Commission Internationale de l'Eclairage. 2017. CIE 2017 colour fidelity index for accurate scientific use. CIE 224:2017. Vienna, Austria: Commission Internationale de l'Eclairage.
- Gardner JL. 2000. Correlated colour temperature - uncertainty and estimation. *Metrologia.* 37(5):381–384. doi:10.1088/0026-1394/37/5/8.
- Grum F, Saunders SB, Macadam DL. 1978. Concept of correlated color temperature. *Color Res Appl.* 3(1):17–21. doi:10.1002/col.5080030107.
- Hernández-Andrés J, Lee RL, Romero J. 1999. Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities. *Appl Opt.* 38(27):5703. doi:10.1364/ao.38.005703.
- [IES] Illuminating Engineering Society. 2020. IES Method for Evaluating Light Source Color Rendition. ANSI/IES TM-30-20. New York, NY: Illuminating Engineering Society.

- [IES] Illuminating Engineering Society. 2020. ANSI/IES LS-1-20: Lighting Science: Nomenclature and Definitions for Illuminating Engineering. New York, NY: Illuminating Engineering Society.
- Krystek M. 1985. An algorithm to calculate correlated colour temperature. *Color Res Appl.* 10(1):38–40.
- Li C, Cui G, Melgosa M, Ruan X, Zhang Y, Ma L, Xiao K, Luo MR. 2016. Accurate method for computing correlated color temperature. *Opt Express.* 24(13):14066. doi:10.1364/oe.24.014066.
- McCamy CS. 1992. Correlated color temperature as an explicit function of chromaticity coordinates. *Color Res Appl.* 17(2):142–144. doi:10.1002/col.5080170211.
- Mori L, Sugiyama H, Kambe N. 1964. An Absolute Method of Color Temperature Measurement. *Acta Chromatica.* 1(3):93–102.
- [NEMA] National Electrical Manufacturers Association. 2017. ANSI/NEMA C78.377-2017 American National Standard for Electric Lamps-Specifications for the Chromaticity of Solid-State Lighting. Products. Rosslyn, Virginia: National Electrical Manufacturers Association.
- Ohno Y. 2014. Practical use and calculation of CCT and Duv. *Leukos.* 10(1):47–55. doi:10.1080/15502724.2014.839020. <http://dx.doi.org/10.1080/15502724.2014.839020>.
- Priest IG. 1933. A proposed scale for use in specifying the chromaticity of incandescent illuminants and various phases of daylight. *JOSA.* 23(2):41–45.
- Prytkov S V., Kolyadin M V. 2021. Error estimation for the methods of correlated colour temperature calculation. *Light Eng.* 29(3):70–77. doi:10.33383/2021-018.
- Robertson AR. 1968. Computation of Correlated Color Temperature and Distribution Temperature. *J Opt Soc Am.* 58(11):1528. doi:10.1364/josa.58.001528.
- Schanda J, Mészáros M, Czibula G. 1978. Calculating correlated color temperature with a desktop programmable calculator. *Color Res Appl.* 3(2):65–69.
- Smet KAG. 2020. Tutorial: The LuxPy Python Toolbox for Lighting and Color Science. *LEUKOS.* 16(3):179–201. doi:10.1080/15502724.2018.1518717.
- Walter W. 1992. Determination of correlated color temperature based on a color-appearance model. *Color Res Appl.* 17(1):24–30. doi:10.1002/col.5080170107.
- Wyszecki G, Stiles WS. 1982. Wyszecki, Gunter, and Walter Stanley Stiles. *Color science. Vol. 8.* New York: Wiley, 1982.
- Xingzhong Q. 1987. Formulas for computing correlated color temperature. *Color Res Appl.* 12(5):285–287.
- Zhang F. 2019. High-accuracy method for calculating correlated color temperature with a lookup table based on golden section search. *Optik (Stuttg).* 193(June):163018. doi:10.1016/j.ijleo.2019.163018.
- Zheleznikova OE, Prytkov SV, Kokinov AM. 2020. Selection of Optimal Method of Correlated Colour Temperature Calculation. *J Mech Contin Math Sci.* spl8(1):134–143. doi:10.26782/jmcms.spl.8/2020.04.00010.