Why You Should Forget Luminance Conversion and Do Something Better

Rang M. H. Nguyen
National University of Singapore
nguyenho@comp.nus.edu.sg

Michael S. Brown
York University
mbrown@eecs.yorku.ca

Abstract

One of the most frequently applied low-level operations in computer vision is the conversion of an RGB camera image into its luminance representation. This is also one of the most incorrectly applied operations. Even our most trusted softwares, Matlab and OpenCV, do not perform luminance conversion correctly. In this paper, we examine the main factors that make proper RGB to luminance conversion difficult, in particular: 1) incorrect white-balance, 2) incorrect gamma/tone-curve correction, and 3) incorrect equations. Our analysis shows errors up to 50% for various colors are not uncommon. As a result, we argue that for most computer vision problems there is no need to attempt luminance conversion; instead, there are better alternatives depending on the task.

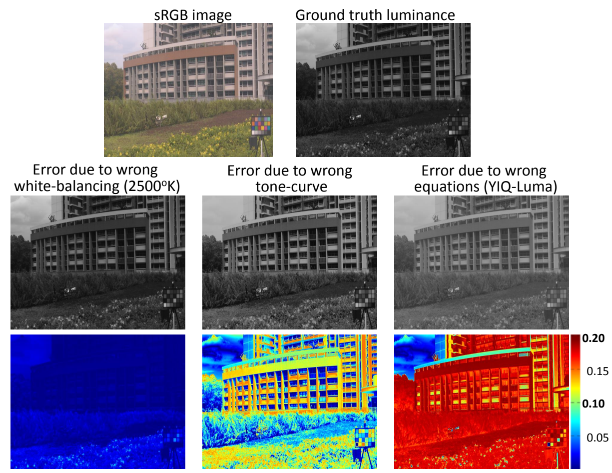

Figure 1. This figure shows examples of errors that arise due to improper luminance conversion. The ground truth luminance for this experiment is captured from a hyperspectral camera.

1. Introduction and Motivation

One of the most frequently applied operations in computer vision and image processing is the conversion of an RGB image into a single-channel luminance representation. Luminance is a photometric measurement that quantifies how the human eye perceives radiant energy emitting from a scene. As such, RGB to luminance conversion is used as a way to convert an RGB image into its perceived *brightness* representation. Luminance is generally represented by the variable, Y, which comes from the CIE 1931 XYZ color space definition for which Y is defined as the luminosity function of a standard human observer under well-lit conditions. Luminance is routinely used in a variety of vision tasks, from image enhancement [22, 27, 29] to feature detection [2, 20], to physical measurements [10, 11, 26].

There are a number of commonly used methods to convert an RGB image to Y. For example, the widely used YIQ and YUV color spaces use the weighted average $Y = 0.299R + 0.587G + 0.114B$, while more recent methods adopt a weighted average of $Y = 0.2126R + 0.7152G + 0.0722B$. In some cases, a simple RGB average of $Y = (R+G+B)/3$ is used. Clearly, these all cannot be correct. In addition, there are other factors at play in this conversion, including the color space's assumed white-point and nonlinear mappings (e.g. gamma correction). Radiometric calibration methods [7, 16, 18, 19] have long known that cameras use proprietary nonlinear mappings (i.e. tone-curves) that do not conform to sRGB standards. Recent work in [3, 15, 17, 31, 14] has shown that these tone-curves can be setting-specific. Fig. 1 shows examples of errors caused by different factors in the color space conversion from sRGB to luminance. Interestingly, however, computer visions algorithms still work in the face of these errors. If our algorithms work with incorrect luminance conversion, why then are we even bothering to attempt luminance conversion?

Contribution This work offers two contributions. First, we systematically examine the challenges in obtaining true scene luminance values from a camera RGB image. Specifically, we discuss the assumptions often overlooked in the definition of standard color spaces and onboard camera photo-finishing that are challenging to undo when performing luminance conversion. We also discuss the use of incorrect equations - e.g YIQ or HSV - that are erroneously interpreted as luminance. Our findings reveal it is not uncommon to encounter conversion errors up to 50% from the true luminance values. Our second contribution is to advocate that for many vision algorithms, alternatives to luminance conversion exist and are better suited for the task at hand.

2. Related Work

There is little work analyzing the correctness of luminance conversion with respect to an imaged scene. Many approaches in the published literature provide a citation to conversion equations given in standard image processing textbooks (e.g., [5]) and assume the conversions to be accurate. There has been work, however, that describes the various color spaces and their usages. Süsstrunk et al. [28] reviewed the specifications and usage of standardized RGB color spaces for images and video. This work described a number of industry-accepted RGB color spaces, such as standard RGB (sRGB), Adobe RGB 98, Apple RGB, NTSC, and PAL. This work serves as a reminder that it is important to be clear about which color space images are in before doing a conversion. Others have examined factors that affect the color distributions of an imaged scene. In particular, Romero et al. [25] analyzed color changes of a scene under variation of daylight illuminations. Their conclusion is that the values of chromaticity coordinates have significant changes while the values of luminance coordinates are less effected. Kanan et al. [13] analyzed the effect of thirteen methods for converting color images to grayscale images (often considered to be luminance) on object recognition. They found, not surprisingly, that different conversion methods result in different object recognition performance.

There is a large body of work on radiometric calibration of cameras (e.g. [7, 16, 18, 19]). These works have long established the importance of understanding the nonlinear mapping of camera pixel intensities with respect to scene radiance. These methods, however, do not explore the relationship of their linearized camera values to the true scene luminance as defined by CIE XYZ.

3. Factors for Luminance Conversion

3.1. Preliminaries: CIE 1931 XYZ and Luminance

Virtually all modern color spaces used in image processing and computer vision trace their definition to the work by Guild and Wright [9, 30], whose work on a device independent perceptual color space was adopted as the official CIE 1931 XYZ color space. Even though other color spaces were introduced later (and shown to be superior), the CIE 1931 XYZ remains the de facto color space for camera and video images.

CIE XYZ (dropping 1931 for brevity) established three hypothetical color primaries, X, Y, and Z. These primaries provide a means to describe a spectral power distribution

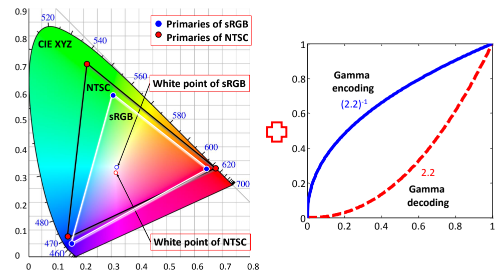

Figure 3. The sRGB and NTSC color spaces primaries and white-points as defined in the CIE XYZ color space. These establish the mapping between CIE XYZ and sRGB/NTSC and vice-versa.

(SPD) by parameterizing it in terms of the X, Y, and Z. This means a three-channel image I under the CIE XYZ color space can be described as:

$$I(\mathbf{x}) = \int_{\omega} C_c(\lambda)\, R(\mathbf{x}, \lambda)\, L(\lambda) d\lambda, \tag{1}$$

where λ represents the wavelength, ω is the visible spectrum $380 - 720$nm, C_c is the CIE XYZ color matching function, and $c = X, Y, Z$ are the primaries. The term $R(\mathbf{x}, \lambda)$ represents the scene's spectral reflectance at pixel $\mathbf{x}$ and $L(\lambda)$ is the spectral illumination in the scene. In many cases, the spectral reflectance and illumination at each pixel are combined together into the spectral power distribution $S(\mathbf{x}, \lambda)$ (see Fig. 2). Therefore, Eq. 1 can be rewritten as:

$$I(\mathbf{x}) = \int_{\omega} C_c(\lambda)\, S(\mathbf{x}, \lambda) d\lambda. \tag{2}$$

In this case, any $S(\mathbf{x})$ that maps to the same X/Y/Z values is considered to be perceived as the *same color* to an observer. The color space was defined such that the matching function associated with the Y primary has the same response as the luminosity function of a standard human observer [4]. This means that the Y value for a given spectral power distribution indicates how bright it is perceived with respect to other scene points. As such, Y is referred to as the "luminance of a scene" and is a desirable attribute to describe an imaged scene.

A number of color spaces have been derived from the CIE XYZ color space. Converting to luminance is essentially mapping a color value in a different color space back to the CIE Y value. The following describes a number of factors necessary to get this mapping correct.

3.2. RGB Color Spaces (sRGB/NTSC)

While CIE XYZ is useful for colorimetry to describe the relationships between SPDs, a color space based on RGB primaries related to real imaging and display hardware is desirable. To establish a new color space, two things are needed - the location of the three primaries (R, G, B) and

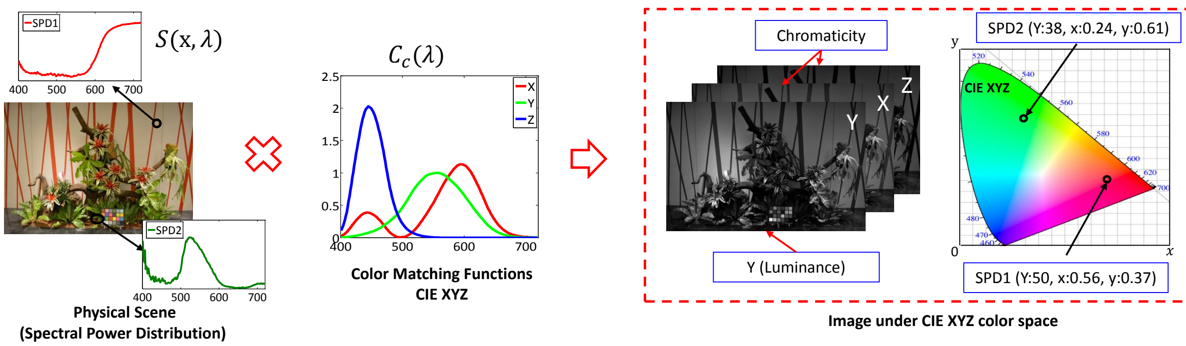

Figure 2. (A) The diagram shows how scene spectral reflectances are converted to the CIE XYZ color space. CIE XYZ proposed three spectral response functions that map real world spectral power distributions (SDPs) to the X/Y/Z basis. The Y value in the CIE XYZ standard is mapped to the standard observer's luminosity function and is taken to represent the perceived brightness of the scene. (B)

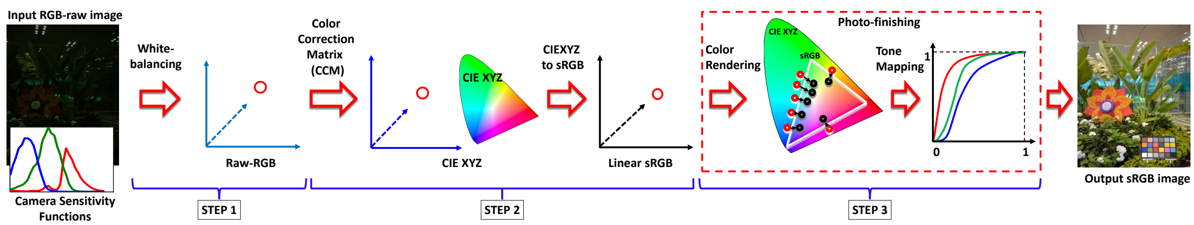

Figure 4. This figure shows the pipeline to obtain sRGB image in consumer cameras. Note that the circles showed in steps 1, 2, and 3 denote for 'white' point while the coordinate systems represent the corresponding color space.

the white-point in CIE XYZ. The white-point is used to determine what CIE XYZ color will represent white (or achromatic colors) in the color space. In particular, it is selected to match the viewing conditions of an image. For example, if it is assumed that a person will be observing a display in daylight, then the CIE XYZ value corresponding to daylight should be mapped to the new color space's white value.

Fig. 3 shows examples for the 1996 sRGB and 1987 National Television System Committee (NTSC) color spaces. Here, NTSC is used as an example. There are many other spaces as noted in [28] - e.g. Adobe RGB, PAL, Apple RGB, and variations over the years, such as NTSC 1953 and NTSC 1987. Each color space has its own 3×3 linear transform based on its respective RGB primaries and white-point location within CIE XYZ.

For the sRGB primaries, the matrix to convert from sRGB to CIE XYZ is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{3}$$

The transform for NTSC (1987) back to CIE XYZ is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.6071 & 0.1736 & 0.1995 \\ 0.2990 & 0.5870 & 0.1140 \\ 0.0000 & 0.0661 & 1.1115 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{4}$$

In both Eqs. 3 and 4, it is important to note that the R, G, B values need to be from images encoded in these respective color spaces. Such R, G, B values are often termed the "linear RGB" values, since both sRGB and NTSC use a final nonlinear gamma function as described in the following.

Gamma sRGB/NTSC were designed for display on CRT monitors and televisions. These devices did not have a linear response to voltage and an *encoding gamma* was applied to the three R/G/B channels as compensation as shown in Fig. 3. For example, a red pixel would take the form $R' = R^{1/\gamma}$, where R is the linear RGB value and R' is the resulting gamma encoded value. This nonlinear gamma was embedded as the final step in the sRGB/NTSC definition. The gamma for NTSC was set to $\gamma = 2.2$; the one for sRGB can be approximated by $\gamma = 2.2$ but is in fact slightly more complicated [1]. Before sRGB or NTSC color spaces can be converted back to CIE XYZ, color values must first be linearized using the inverse gamma.

3.3. Camera Imaging Pipeline

The vast majority of consumer cameras save their images in the sRGB color space. In an ideal scenario, luminance can be computed by first applying an inverse gamma followed by:

$$Y = 0.2126R + 0.7152G + 0.0722B, \tag{5}$$

which comes directly from Eq. 3. However, while images are encoded using sRGB, virtually all camera image pipelines deviate from the sRGB standard. Fig. 4 shows an overview of the common steps in a camera image pipeline. First, the camera sensitivity of a camera sensor is not the same as CIE XYZ. This means that camera images are in their own raw-RGB color space, which must be converted to sRGB [14]. Before this happens, the image is generally white-balanced using a diagonal 3×3 matrix to remove illumination color casts and properly map the scene's white colors to lie along the achromatic line (i.e., $R = G = B$). After white-balancing, the image's raw-RGB values are converted to CIE XYZ using a 3×3 color correction matrix (CCM). Once in the CIE XYZ color space, the image can be mapped to linear-sRGB and the sRGB encoding gamma is applied. However, most cameras apply their own tone-curve [7, 16, 18, 19] and/or additional selective color rendering [3, 15, 17, 31] as part of their proprietary photo-finishing.

Examining the pipeline, we can see that there are two factors that can affect luminance conversion. If white-balancing is applied before the CIE XYZ conversion, an incorrect white-point estimation can cause errors when the CCM is applied. Next, if the tone-curve deviates strongly from the sRGB encoding gamma it will introduce errors in the linearization step when converting back from sRGB to CIE XYZ.

3.4. Incorrect Y Conversion and Luma

As previously discussed, if a camera manufacturer carefully follows the sRGB standard, the gamma decoded linear-sRGB can be converted back to Y by using Eq. 5. However, it is often the case that completely incorrect conversion methods are used. The following are three incorrect methods commonly found in the computer vision literature.

The first is to compute the average RGB values (which is typically applied to nonlinear RGB images). This is computed as:

$$Y = \frac{1}{3}(R + G + B). \tag{6}$$

It is curious to wonder why this would be considered luminance, but as we will show in Sec. 4, this is not a bad choice.

Another commonly applied conversion from sRGB is to the YIQ or YUV color spaces [5]. YIQ and YUV are derived from the NTSC 1987 color space, and are technically defined on the gamma encoded R, G, B values. These color spaces should be denoted as Y′IQ or Y′UV, where the prime symbol is used to distinguish Y′ from Y that represents luminance. In the video engineering community, the term 'luma' is also used to refer to Y′ and is not intended to represent luminance. As noted by Poynton [24] this distinction of luma has been confused in the image processing and graphics community and is incorrectly interpreted as luminance. The incorrect luminance conversion is derived from Eq. 4 as follows:

$$Y = 0.299R + 0.587G + 0.114B. \tag{7}$$

This equation is the most commonly applied conversion to luminance found in the academic literature, however, it is almost always preformed incorrectly. Only if the image is captured in the NTSC color space and the proper gamma decoding has been applied, then it is a valid conversion. If used with linear-sRGB images (which most modern cameras use), this equation attempt sto convert from the wrong color space, because the transform is based on different RGB primaries and white-point related to NTSC and not sRGB. When no gamma decoding is applied, it converts to luma based of NTSC. There are a number of well-known methods that use this conversion, including Matlab's `rgb2gray` function and OpenCV's `cv2.cvtColor` function.

Another common conversion often confused with luminance is the "value" definition in the hue, saturation, value (HSV) color space [5]. HSV defines *value* as the maximum value of the R/G/B channel for each pixel:

$$Y = \max(R, G, B). \tag{8}$$

As with Eq. 6, the relationship of this conversion to scene luminance is unclear.

Luminance vs. Luma As previously mentioned, when applying a conversion on the gamma encoded R,G,B values, the result should *not* be called luminance, but is instead referred to as luma. While certain spaces, e.g. YUV and YIQ, are defined on the gamma encoded RGB values (and should technically be written with Y'), a common practice in the literature is not to perform the gamma decoding step when doing a conversion. In this paper, we distinguish this by adding the term 'Luma' to the conversion type – e.g. YIQ-Luma or sRGB-Luma.

4. Luminance Conversion Analysis

Sec. 3 has discussed the proper sRGB to luminance conversion and where potential errors can occur either in the camera imaging pipeline or due to incorrect conversion methods. The goal of this section is to examine the effect of each factor, in particular: white-balance, tone-curve, and erroneous conversion methods (YIQ, HSV, averageRGB).

To establish the ground truth luminance for a scene, we use Specim's PFD-CL-65-V10E hyperspectral camera to capture the spectral power distributions of several real scenes as well as a 140-patch Macbeth color checker pattern. This allows us to compute the ground truth luminance by applying the CIE XYZ matching functions directly to the spectral scene to obtain Y. Our experiments are performed on synthetic images that allow us to carefully control the pipeline and on real camera images as described in the following section.

4.1. Computing Synthetic Camera Images

To be able to control various components of the camera image pipeline, we synthesize sRGB images for the following two cameras: 1) a Canon 1Ds Mark III and 2) a Nikon D40. We do this by emulating the camera processing pipeline as described in Fig. 4. The sensor sensitivity functions for these cameras were estimated in the work by Jiang et al. [12]. This allows us to synthesize a camera's raw-RGB by applying the associated camera sensitivity functions to the spectral images. Next, we apply white-balance on the raw-RGB images (the correct white is known from white patches placed in the scene). After this, the CCM computed using the methods proposed in [23] is applied to convert the raw-RGB to CIE XYZ. Finally, the final sRGB can be computed using either the correct encoding gamma function (2.2) or the estimated tone-curves of these cameras available from [17]. Note that all errors are reported in normalized pixel values between [0-1].

4.2. White-Balance with Proper Gamma

Our first experiment examines white-balance's effect on luminance conversion. We generate synthetic images as described in Sec. 4.1 but purposely use the incorrect color temperatures to white-balance the image - namely 2500K, 4000K, and 10000K (the correct white-balance is at 6000K). To isolate the errors to white-balance, we use the proper sRGB encoding gamma of 2.2.

Fig. 5 shows the quantitative luminance error for the color chart between the ideal white-balanced image and incorrectly white-balanced images. A jet map is used to highlight the error between the ground truth and estimated luminance. We show the quantitative error statistics: maximum (Max), mean (Mean), and standard deviation (Std). It is clear that the worse white-balancing (2500K) results in more error. However, the overall errors are not that significant, around 1% on average for the worst case.

4.3. Tone-Curve

Our next experiment examines the effect of the camera's tone-curves. The proper white-balance is applied; however, instead of the 2.2 sRGB gamma mapping, we used the camera-specific tone-curves from [17]. However, when we linearize the sRGB image, we use the known 2.2 decoding gamma. Fig. 6 shows the quantitative error of the luminance channel for two different cameras. The improper linearization causes significant errors, ranging from 10% to 18% depending on the camera.

4.4. Wrong Luminance Conversion

In this experiment, we examined the effect of using the incorrect conversion methods discussed in Sec. 3.4 - YIQ, HSV and RGB average. Images are rendered with the proper white-balance and an encoding gamma of 2.2. This means the input images are as close to ideal sRGB as possible. We apply these approaches using the proper sRGB decoding gamma and without any linearization - i.e. we compute the incorrect "luma". Although YIQ is defined on the gamma encoded RGB space, we show results with and without gamma decoding applied, this is denoted as YIQ and YIQ-Luma respectively.

	Canon 1Ds Mark III			**Nikon D40**		
	Max	**Mean**	**Std**	**Max**	**Mean**	**Std**
YIQ	0.048	0.018	0.011	0.048	0.018	0.010
1/3	0.127	0.025	0.028	0.126	0.027	0.027
HSV	0.287	0.045	0.053	0.281	0.048	0.053
YIQ-Luma	0.332	0.252	0.083	0.332	0.253	0.083
1/3-Luma	0.349	0.230	0.093	0.350	0.232	0.094
HSV-Luma	0.478	0.271	0.106	0.477	0.274	0.108

Table 1. [Color chart] This table shows quantitative error for the synthetic images of the color chart using camera sensitivity functions of two different cameras, Canon 1D and Nikon D40 in [12]. An encoding gamma of 2.2 is applied to synthesize the sRGB images.

	Canon 1Ds Mark III			**Nikon D40**		
	Max	**Mean**	**Std**	**Max**	**Mean**	**Std**
YIQ	0.077	0.004	0.003	0.076	0.003	0.003
1/3	0.165	0.002	0.002	0.174	0.002	0.002
HSV	0.373	0.024	0.020	0.447	0.023	0.020
YIQ-Luma	0.327	0.224	0.068	0.328	0.222	0.068
1/3-Luma	0.384	0.214	0.063	0.391	0.212	0.063
HSV-Luma	0.577	0.251	0.082	0.626	0.250	0.081

Table 2. [Outdoor scene] This table shows quantitative error for synthetic images of an outdoor scene using camera sensitivity functions of two different cameras Canon 1D and Nikon D40 in [12]. An encoding gamma of 2.2 is applied to synthesize the sRGB images.

Tabs. 1 and 2 show the quantitative error for the color checker chart and an outdoor scene respectively. The tables reveal that improper conversion (with linearization) results in errors ranging from 1% to 5% for two different cameras. The outdoor scene (shown in Fig. 7) is not as bad, but contains less color variation than the color chart. The estimation of luma, however, results in significant errors, with average errors ranging from 30% to over 50%.

4.5. Real Camera Images

Our final experiment uses images captured from cameras that were placed next to our spectral camera. The images have been carefully aligned to the color chart image using a homography. These real images are captured by the same models used in our synthetic experiments. Fig. 8 shows the quantitative error between the luminance synthesized by CIE XYZ color matching functions (ground truth) and the real sRGB images from the Canon 1D camera. The top row shows the comparison between ground truth lumi-

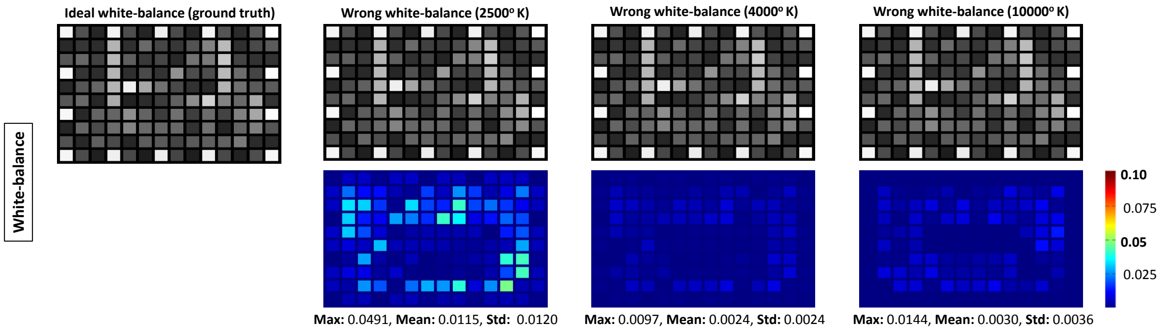

Figure 5. This figure shows the quantitative error of improper white-balance on the luminance channel. The white-balance matrices of three incorrect color temperatures - 2500K, 4000K, and 10000K (the correct white-balance is at 6000K) - are applied on the color chart image.

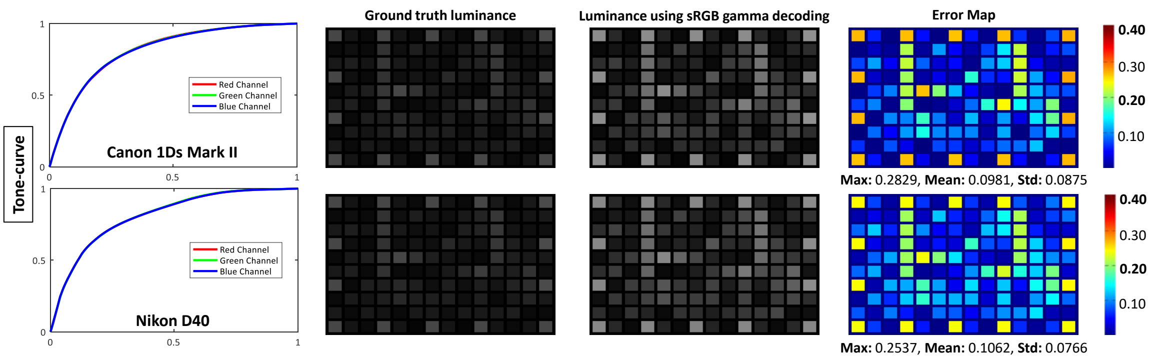

Figure 6. This figure shows the errors that occur when the camera's true tone-curve is not used to linearize the sRGB values. Errors range from 10% to 18%.

	Canon 1Ds Mark III			Nikon D40		
	Max	**Mean**	**Std**	**Max**	**Mean**	**Std**
YIQ-T	0.137	0.056	0.032	0.078	0.027	0.020
1/3-T	0.211	0.065	0.042	0.171	0.041	0.036
HSV-T	0.382	0.090	0.077	0.532	0.111	0.120
YIQ-G	0.227	0.123	0.054	0.200	0.114	0.043
1/3-G	0.288	0.125	0.063	0.276	0.114	0.057
HSV-G	0.357	0.111	0.067	0.523	0.124	0.087
YIQ-Luma	0.181	0.068	0.047	0.153	0.056	0.037
1/3-Luma	0.222	0.071	0.053	0.213	0.064	0.045
HSV-Luma	0.501	0.159	0.123	0.540	0.168	0.141

Table 3. This table shows quantitative error for the real images of color chart captured by two different cameras, Canon 1D and Nikon D40. T denotes the proper tone-curve of the camera and G denotes the gamma of 2.2.

nance and the luminance from the linearized sRGB using sRGB gamma correction. The bottom row shows the comparison between the ground truth luminance and the luminance from the linearized sRGB using the camera's tone-curve measured in [17]. The results show that it is very important to use the correct tone-curves to linearize the RGB color values before computing luminance. It is also worth noting that in cases of using the correct tone-curve, computing luminance values still has a small error (around 2%) that is caused by factors such as the inaccuracy of white-balancing, the CCM accuracy, and the selective color manipulation noted in [17, 15, 31].

We also examined the different conversion methods with different linearization, in particular the proper tone-curves (T), the decoding gamma of 2.2 (G), and without linearization. Tab. 3 shows the quantitative errors that occur. The results show that using the proper tone-curves have the least amount of error on computing luminance. Interestingly, using the recommended decoding gamma of 2.2 can lead to more error than without linearization.

5. Better Alternatives

Sec. 4 showed that luminance conversion is prone to significant error. Our analysis is not intended to imply that prior work using these methods are incorrect, or that their results would be better if they were able to estimate true scene luminance. What our results implies is that a many prior works are incorrectly interpreting what is truly being processed by their methods. In fact, for many applications, there is often no reason to attempt a luminance conversion. In this section, we discuss alternatives to luminance conversion for two well-known tasks, tone-mapping and feature detection.

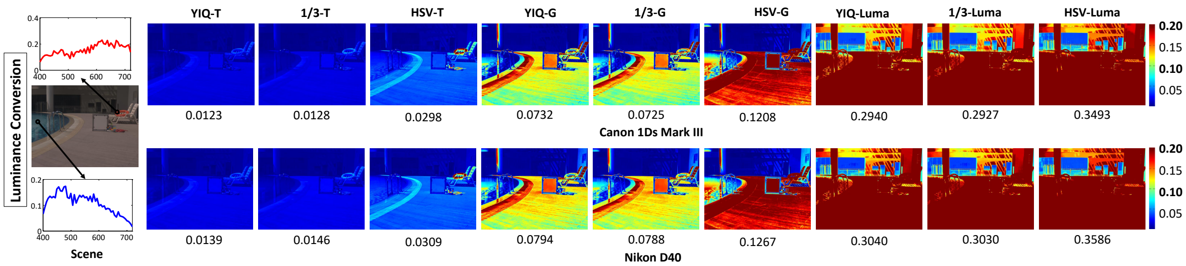

Figure 7. This figure shows qualitative error for the synthetic images of an outdoor scene using camera sensitivity functions of two different cameras Canon 1D and Nikon D40 in [12]. A gamma of 2.2 is applied to obtain the sRGB images.

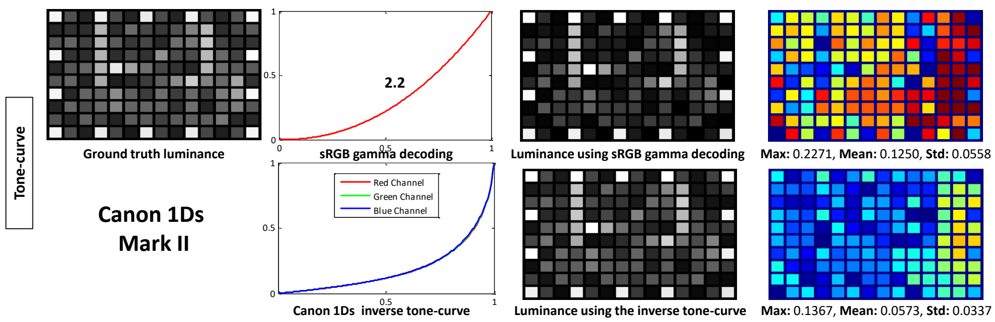

Figure 8. This figure shows the quantitative error between the luminance synthesized by CIE XYZ color matching functions (ground truth) and real sRGB image from the camera Canon 1D. The top row shows the comparison between ground truth luminance and the luminance from the linearized sRGB using sRGB gamma correction. The bottom row shows the comparison between ground truth luminance and the luminance from the linearized sRGB using the camera's tone-curve measured in [17].

5.1. Tone-Mapping

Tone-mapping applies a nonlinear scaling to an image's intensity values to enhance contrast. This is often performed by first decomposing an image to its luminance representation, processing the luminance channel, and then converting back. However, as we have shown, most luminance conversions are incorrect, making it hard to interpret what is truly being processed.

Instead, what is generally desired in tone-mapping is to modify contrast while maintaining chromaticity in the current RGB color space. This means that after processing, neutral colors will stay "white", and the chromaticity of other colors will also not be changed. We examine three methods: YIQ, HSV, and *average*-RGB (1/3-RGB), to see their ability to preserve chromaticity after tone-mapping. For this experiment, an image of a color chart is transformed by each method, its corresponding "luminance" is stretched by the same nonlinear tone-curve, and the new RGB image is obtained by transforming back. Note that the *average*-RGB defines only a single-channel; how to use this and transform back to RGB is explained in the supplemental material. Fig. 9 shows the effect of tone-mapping operator on the color chromaticity of these methods. The red points show the original chromaticities while the blue ones show the chromaticities after tone-mapping. For the neutral colors, all four methods can preserve their chromaticities.

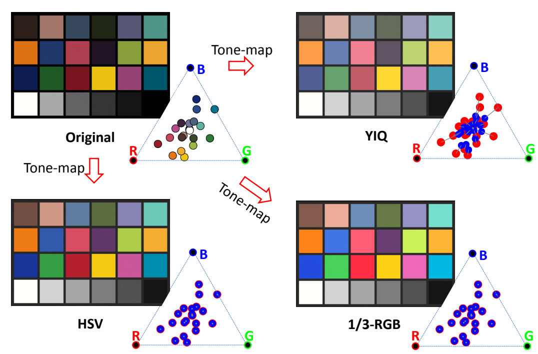

Figure 9. This figure shows the change of the color chromaticity of four different methods on tone mapping operator. The red points show the original chromaticities while the blue ones show the chromaticities after tone-mapping.

However, only HSV and 1/3-RGB can preserve the chromaticity of chromatic colors. The color chromaticities in the case of YIQ are shifted towards the achromatic point (e.g. $R = G = B$). Interestingly, attempting luminance conversion gives the undesired effect, while non-luminance methods satisfy the goal of this task.

5.2. Feature Detection

Converting to luminance for feature detection is often used to reduce the RGB channels to a single image for faster

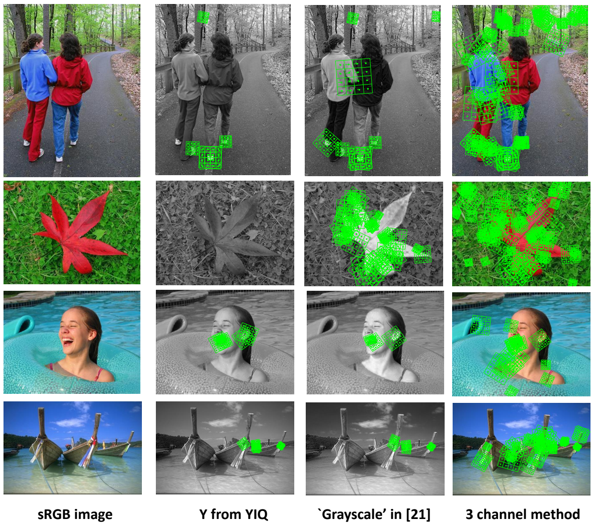

Figure 10. This figure shows several examples of SIFT feature detection. The first column is the color input images. The second and third columns show the results of using Y channel from YIQ and grayscale images obtained from [21], respectively. The last column shows the results by applying SIFT feature detection on three color channels and combining them together.

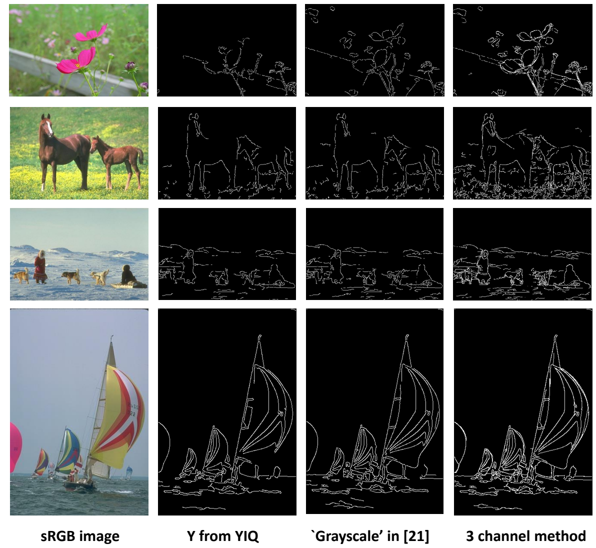

Figure 11. This figure shows several examples of Canny edge detection. The first column is the color input images. The second and third columns show the results of using Y channel from YIQ and grayscale images obtained from [21], respectively. The last column shows the results by applying Canny edge detection on three color channels and combining them together.

processing. Here, we experiment with two common features, SIFT [20] and Canny edge detection [2]. Both these methods aim to detect useful image features by examining image gradient caused by scene object texture or boundaries. When single-channel processing is desired, it is not important to obtain true scene luminance, but a representation that maintains gradients between scene objects and texture. To this end, color-to-gray methods (e.g. [6, 8, 21]) that convert an RGB to a gray image while preserving some notion of RGB contrast are useful. Figs. 10 and 11 show several examples for Canny edge and SIFT detection on three different methods: using the Y channel from YIQ, the saliency-preserving decolorization [21], and processing all three color channels independently and then aggregating the results. As can be seen, the color-to-gray conversion method helps to preserve the color contrast and allows SIFT and Canny to obtain better features than the simple conversion Y of YIQ. When faster processing is not needed, processing all three color channels independently and aggregating the results often give the best performance. See additional results in our supplemental material.

6. Discussion and Summary

This paper provided analysis of one of the most common yet incorrectly applied operations used in computer vision and image processing applications - conversion of a camera RGB image to a scene luminance representation. To the best of our knowledge, this is the first paper to systematically examine the various factors that lead to errors, in particular - 1) incorrect equations, 2) incorrect white-balance, and 3) improper gamma/tone-curve correction.

Our analysis is not intended to suggest that existing computer vision methods that apply a luminance conversion are incorrect, or that existing algorithms would benefit from a more accurate luminance conversion. Instead, our work serves to justify alternative conversion methods that are not based on color science for use in converting an sRGB camera image to a single channel representation. In fact, the vast majority of computer vision algorithms do not rely on the colorimetric properties of the RGB signals, but instead rely on cues in terms of signal difference (gradient) that can be obtained from a number of different types of single-channel representations. Our hope is that this work will motivate researchers not to feel compelled to apply an erroneous luminance conversion in order to appear scientifically justified when alternatives, such as simple *average*-RGB, are just as valid and potentially work better.

Acknowledgements

This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme and a Google Research Award.

References

[1] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes. Proposal for a standard default color space for the internet - srgb. In *Color and Imaging Conference*, pages 238–245, 1996. 3

[2] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986. 1, 8

[3] A. Chakrabarti, Y. Xiong, B. Sun, T. Darrell, D. Scharstein, T. Zickler, and K. Saenko. Modeling radiometric uncertainty for vision with tone-mapped color images. *TPAMI*, 36(11):2185–2198, 2014. 1, 4

[4] H. S. Fairman, M. H. Brill, and H. Hemmendinger. How the CIE 1931 color-matching functions were derived from Wright-Guild data. *Color Research & Application*, 22(1):11–23, 1997. 2

[5] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Prentice Hall Upper Saddle River, NJ:, 2002. 2, 4

[6] A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch. Color2gray: salience-preserving color removal. *ACM Transactions on Graphics (TOG)*, 24(3):634–639, 2005. 8

[7] M. Grossberg and S. Nayar. What is the Space of Camera Response Functions? In *CVPR*, pages 602–609, 2003. 1, 2, 4

[8] M. Grundland and N. A. Dodgson. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recognition*, 40(11):2891–2896, 2007. 8

[9] J. Guild. The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society of London*, pages 149–187, 1932. 2

[10] H. Haussecker and D. Fleet. Computing optical flow with physical models of brightness variation. In *CVPR*, pages 760–767, 2000. 1

[11] K. Ikeuchi and B. K. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1):141–184, 1981. 1

[12] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk. What is the space of spectral sensitivity functions for digital color cameras? In *WACV*, pages 168–179, 2013. 5, 7

[13] C. Kanan and G. W. Cottrell. Color-to-grayscale: does the method matter in image recognition? *PloS ONE*, 7(1), 2012. 2

[14] H. Karaimer and M. S. Brown. A software platform for manipulating the camera imaging pipeline. In *ECCV*, pages 429–444, 2016. 1, 4

[15] S. J. Kim, H. T. Lin, Z. Lu, S. Süsstrunk, S. Lin, and M. S. Brown. A new in-camera imaging model for color computer vision and its application. *TPAMI*, 34(12):2289–2302, Dec 2012. 1, 4, 6

[16] S. J. Kim and M. Pollefeys. Robust radiometric calibration and vignetting correction. *TPAMI*, 30(4):562–576, 2008. 1, 2, 4

[17] H. Lin, S. J. Kim, S. Süsstrunk, and M. S. Brown. Revisiting radiometric calibration for color computer vision. In *ICCV*, pages 129–136, 2011. 1, 4, 5, 6, 7

[18] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric calibration from a single image. In *CVPR*, pages 938–945, 2004. 1, 2, 4

[19] S. Lin and L. Zhang. Determining the radiometric response function from a single grayscale image. In *CVPR*, pages 66–73, 2005. 1, 2, 4

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 8

[21] C. Lu, L. Xu, and J. Jia. Contrast preserving decolorization with perception-based quality metrics. *IJCV*, 110(2):222–239, 2014. 8

[22] S. K. Naik and C. Murthy. Hue-preserving color image enhancement without gamut problem. *TIP*, 12(12):1591–1598, 2003. 1

[23] R. M. H. Nguyen, D. K. Prasad, and M. S. Brown. Raw-to-raw: Mapping between image sensor color responses. In *CVPR*, pages 3398–3405, 2014. 5

[24] C. Poynton. *Digital Video and HDTV: Algorithms and Interfaces*. Morgan Kaufmann, 2012. 4

[25] J. Romero, J. Hernández-Andrés, J. L. Nieves, and J. A. García. Color coordinates of objects with daylight changes. *Color Research & Application*, 28(1):25–35, 2003. 2

[26] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *ICCV*, pages 875–882, 1999. 1

[27] J.-L. Starck, F. Murtagh, E. Candes, and D. Donoho. Gray and color image contrast enhancement by the curvelet transform. *TIP*, 12(6):706–717, 2003. 1

[28] S. Süsstrunk, R. Buckley, and S. Swen. Standard rgb color spaces. In *Color and Imaging Conference*, pages 127–134, 1999. 2, 3

[29] A. Toet. Multiscale color image enhancement. *Pattern Recognition Letters*, 13(3):167–174, 1992. 1

[30] W. D. Wright. A re-determination of the trichromatic coefficients of the spectral colours. *Transactions of the Optical Society*, 30(4):141, 1929. 2

[31] Y. Xiong, K. Saenko, T. Darrell, and T. Zickler. From pixels to physics: Probabilistic color de-rendering. In *CVPR*, pages 358–365, 2012. 1, 4, 6