

Picture Perfect *RGB* Rendering Using Spectral Prefiltering and Sharp Color Primaries

Greg Ward[†]

Elena Eydelberg-Vileshin[‡]

Abstract

Accurate color rendering requires the consideration of many samples over the visible spectrum, and advanced rendering tools developed by the research community offer multispectral sampling towards this goal. However, for practical reasons including efficiency, white balance, and data demands, most commercial rendering packages still employ a naive RGB model in their lighting calculations. This results in colors that are often qualitatively different from the correct ones. In this paper, we demonstrate two independent and complementary techniques for improving RGB rendering accuracy without impacting calculation time: spectral prefiltering and color space selection. Spectral prefiltering is an obvious but overlooked method of preparing input colors for a conventional RGB rendering calculation, which achieves exact results for the direct component, and very accurate results for the interreflected component when compared with full-spectral rendering. In an empirical error analysis of our method, we show how the choice of rendering color space also affects final image accuracy, independent of prefiltering. Specifically, we demonstrate the merits of a particular transform that has emerged from the color research community as the best performer in computing white point adaptation under changing illuminants: the Sharp RGB space.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Color, shading, shadowing, and texture

1. Introduction

It is well-known that the human eye perceives color in a three-dimensional space, owing to the presence of three types of color receptors. Early psychophysical research demonstrated conclusively that three component values are sufficient to represent any perceived color, and these values may be quantified using the CIE *XYZ* tristimulus space²⁰. However, because the spectrum of light is continuous, the interaction between illumination and materials cannot be accurately simulated with only three samples. In fact, no finite number of fixed spectral samples is guaranteed to be sufficient — one can easily find pathological cases, for example, a pure spectral source mixed with a narrow band absorber, that require either component analysis or a ludicrous number of fixed samples to resolve. If the rendered spectrum is

inaccurate, reducing it to a tristimulus value will usually not hide the problem.

Besides the open question of how many spectral samples to use, there are other practical barriers to applying full spectral rendering in commercial software. First, there is the general dearth of spectral reflectance data on which to base a spectral simulation. This is consistent with the lack of any kind of reflectance data for rendering. We are grateful to the researchers who are hard at work making spectral data available^{3,19}, but the ultimate solution may be to put the necessary measurement tools in the hands of people who care about accurate color rendering. Hand-held spectrophotometers exist and may be purchased for the cost of a good laser printer, but few people apply them in a rendering context, and to our knowledge, no commercial rendering application takes spectrophotometer data as input.

The second practical barrier to spectral rendering is white balance. This is actually a minor issue once you know how to address it, but the first time you render with the correct

[†] Exponent – Failure Analysis Associates, Menlo Park, California

[‡] Department of Computer Science, Stanford University, Palo Alto, California

source and reflectance spectra, you are likely to be disappointed by the strong color cast in your output. This is due to the change in illuminant from the simulated scene to the viewing condition, and there is a well-known method to correct for this, which we will cover in Section 2.

The third practical barrier to the widespread acceptance of spectral rendering is what we call the “data mixing problem.” What if the user goes to the trouble of acquiring spectral reflectances for a set of surfaces, but they also want to include materials that are characterized in terms of *RGB* color, or light sources that are specified to a different spectral resolution? One may interpolate and extrapolate to some extent, but in the end, it may be necessary to either synthesize a spectrum from *RGB* triples a la Smits’ method¹⁴, or reduce all the spectral data to *RGB* values and fall back on three component rendering again.

The fourth practical barrier to full spectral rendering is cost. In many renderings, shading calculations dominate the computation, even in *RGB*. If all of these calculations must be carried out at the maximum spectral resolution of the input, the added cost may not be worth the added benefit.

Many researchers in computer graphics and color science have addressed the problem of efficient spectral sampling^{8,7}. Meyer suggested a point-sampling method based on Gaussian quadrature and a preferred color space, which requires only 4 spectral samples and is thus very efficient¹. Like other point sampling techniques, however, Meyer’s method is prone to problems when the source spectrum has significant spikes in it, as in the case of common fluorescent lighting. A more sophisticated approach employing orthonormal basis functions was presented by Peercy, who uses characteristic vector analysis on combinations of light source and reflectance spectra to find an optimal, orthonormal basis set¹³. Peercy’s method has the advantage of handling spiked and smooth spectra with equal efficiency, and he demonstrated accurate results with as few as three orthonormal bases. The additional cost is comparable to spectral sampling, replacing N multiplies in an N -sample spectral model with $M \times M$ multiplies in an M -basis vector model. Examples in his paper showed the method significantly out-performing uniform spectral sampling for the same number of operations. The cost for a 3-basis simulation, the minimum for acceptable accuracy in Peercy’s technique, is roughly three times that of a standard *RGB* shading calculation.

In this paper, we present a method that has the same overall accuracy as Peercy’s technique, but without the computational overhead. In fact, no modification at all is required to a conventional *RGB* rendering engine, which multiplies and sums its three color components separately throughout the calculation. Our method is not subject to point sampling problems in spiked source or absorption spectra, and the use of an *RGB* rendering space all but eliminates the data mixing problem mentioned earlier. White adaptation is also accounted for by our technique, since we ask the user to iden-

tify a dominant source spectrum for their scene. This avoids the dreaded color cast in the final image.

We start with a few simple observations:

1. The direct lighting component is the first order in any rendering calculation, and its accuracy determines the accuracy of what follows.
2. Most scenes contain a single dominant illuminant; there may be many light sources, but they tend to all have the same spectral power distribution, and spectrally differentiated sources make a negligible contribution to illumination.
3. Exceptional scenes, where spectrally distinct sources make roughly equal contributions, cannot be “white balanced,” and will look wrong no matter how accurately the colors are simulated. We can be satisfied if our color accuracy is no worse on average than standard methods in the mixed illuminant case.

The spectral prefiltering method we propose is quite simple. We apply a standard CIE formula to compute the reflected *XYZ* color of each surface under the dominant illuminant, then transform this to a white-balanced *RGB* color space for rendering and display. The dominant sources are then replaced by white sources of equal intensity, and other source colors are modified to account for this adaptation. By construction, the renderer gets the exact answer for the dominant direct component, and a reasonably close approximation for other sources and higher order components.

The accuracy of indirect contributions and spectrally distinct illumination will depend on the sources, materials, and geometry in the scene, as well as the color space chosen for rendering. We show by empirical example how a sharpened *RGB* color space seems to perform particularly well in simulation, and offer some speculation as to why this might be the case.

Section 2 details the equations and steps needed for spectral filtering and white point adjustment. Section 3 shows an example scene with three combinations of two spectrally distinct light sources, and we compare the color accuracy of naive *RGB* rendering to our prefiltering approach, each measured against a full spectral reference solution. We also look at three different color spaces for rendering: CIE *XYZ*, linear *sRGB*, and the Sharp *RGB* space. Finally, we conclude with a summary discussion and suggestions for future work.

2. Method

The spectral prefiltering method we propose is a straightforward transformation from measured source and reflectance spectra to three separate color channels for rendering. These input colors are then used in a conventional rendering process, followed by a final transformation into the display *RGB* space. Chromatic adaptation (i.e., white balancing) may take place either before or after rendering, as a matter of convenience and efficiency.

2.1. Color Transformation

Given a source $I(\lambda)$ and a material $\rho_m(\lambda)$ with arbitrary spectral distributions, the CIE describes a standard method for deriving a tristimulus value that quantifies the average person's color response. The XYZ tristimulus color space is computed from the CIE "standard observer" response functions, \bar{x} , \bar{y} , and \bar{z} , which are integrated with an arbitrary source illuminant spectrum and surface reflectance spectrum as shown in Eq. (1), below:

$$\begin{aligned} X_m &= \int I(\lambda) \rho_m(\lambda) \bar{x}(\lambda) d\lambda \\ Y_m &= \int I(\lambda) \rho_m(\lambda) \bar{y}(\lambda) d\lambda \\ Z_m &= \int I(\lambda) \rho_m(\lambda) \bar{z}(\lambda) d\lambda \end{aligned} \quad (1)$$

For most applications, the 1971 2° standard observer curves are used, and these may be found in Wyszecki and Stiles²⁰.

Eq. (1) is very useful for determining metameric color matches, but it does not give us an absolute scale for color appearance. For example, there is a strong tendency for viewers to discount the illuminant in their observations, and the color one sees depends strongly on the ambient lighting and the surround. For example, Eq. (1) might compute a yellow-orange color for a white patch under a tungsten illuminant, while a human observer would still call it "white" if they were in a room lit by the same tungsten source. In fact, a standard photograph of the patch would show its true yellow-orange color, and most novice photographers have the experience of being startled when the colors they get back from their indoor snapshots are not as they remembered them.

To provide for the viewer's chromatic adaptation and thus avoid a color cast in our image after all our hard work, we apply a von Kries style linear transform to our values prior to display¹⁷. This transform takes an XYZ material color computed under our scene illuminant, and shifts it to the equivalent, apparent color XYZ' under a different illuminant that corresponds to our display viewing condition. All we need are the XYZ colors for white under the two illuminants as computed by Eq. (1) with $\rho_m(\lambda) = 1$, and a 3×3 transformation matrix, M_C , that takes us from XYZ to an appropriate color space for chromatic adaptation. (We will discuss the choice of M_C shortly.) The combined adaptation and display transform is given in Eq. (2), below:

$$\begin{bmatrix} R'_m \\ G'_m \\ B'_m \end{bmatrix} = M_D M_C^{-1} \begin{bmatrix} R'_w & 0 & 0 \\ 0 & G'_w & 0 \\ 0 & 0 & B'_w \end{bmatrix} M_C \begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix}, \quad (2)$$

where

$$\begin{bmatrix} R_w \\ G_w \\ B_w \end{bmatrix} = M_C \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

for the scene illuminant, and similarly for the display white point, (X'_w, Y'_w, Z'_w) .

The display matrix, M_D , that we added to the standard von Kries transform, takes us from CIE XYZ coordinates to our display color space. For an $sRGB$ image or monitor with D65 white point¹⁵, one would use the following matrix, followed by a gamma correction of 1/2.2:

$$M_{sRGB} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix}$$

If we are rendering a high dynamic-range scene, we may need to apply a tone-mapping operator such as Larson et al⁶ to compress our values into a displayable range. The tone operator of Pattanaik et al even incorporates a partial chromatic adaptation model¹².

The choice of which matrix to use for chromatic adaptation, M_C , is an interesting one. Much debate has gone on in the color science community over the past few years as to which space is most appropriate, and several contenders seem to perform equally well in side-by-side experiments². However, it seems clear that RGB primary sets that are "sharper" (more saturated) tend to be more plausible than primaries that are inward of the spectral locus⁴. In this paper, we have selected the *Sharp* adaptation matrix for M_C , which was proposed based on spectral sharpening of color-matching data¹⁷:

$$M_{Sharp} = \begin{bmatrix} 1.2694 & -0.0988 & -0.1706 \\ -0.8364 & 1.8006 & 0.0357 \\ 0.0297 & -0.0315 & 1.0018 \end{bmatrix}$$

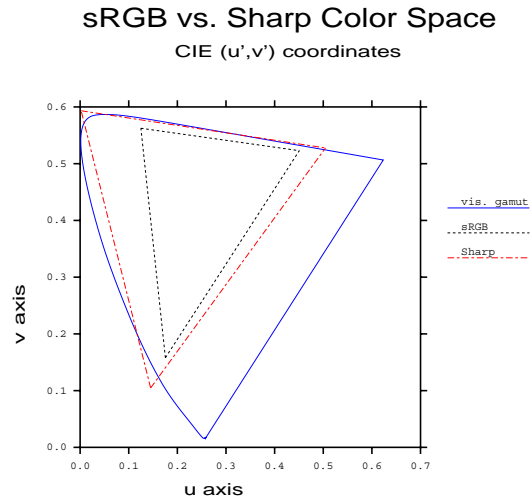


Figure 1: A plot showing the relative gamuts of the $sRGB$ and Sharp color spaces.

Figure 1 shows a CIE (u', v') plot with the locations of the *sRGB* and Sharp color primaries relative to the visible gamut. Clearly, one could not manufacture a color monitor with Sharp primaries, as they lie just outside the spectral locus. However, this poses no problem for a color transform or a rendering calculation, since we can always transform back to a displayable color space.

In fact, the Sharp primaries may be preferred for rendering and *RGB* image representation simply because they include a larger gamut than the standard *sRGB* primaries. This is not an issue if one can represent color values less than zero and greater than one, but most image formats and some rendering frameworks do not permit this. As we will see in Section 3, the choice of color space plays a significant role in the final image accuracy, even when gamut is not an issue.

2.2. Application to Rendering

We begin with the assumption that the direct-diffuse component is most important to color and overall rendering accuracy. Inside the shader of a conventional *RGB* rendering system, the direct-diffuse component is computed by multiplying the light source color by the diffuse material color, where color multiplication happens separately for each of the three *RGB* values. If this calculation is accurate, it must give the same result one would get using Eq. (1) followed by conversion to the rendering color space. In general, this will not be the case, because the diffuse *RGB* for the surface will be based on some other illuminant whose spectrum does not match the one in the model.

For example, the CIE (x, y) chromaticities and *Y*-reflectances published on the back of the Macbeth ColorChecker chart⁹ are measured under standard illuminant *C*, which is a simulated overcast sky. If a user wants to use the Macbeth color Purple in his *RGB* rendering of an interior space with an incandescent (tungsten) light source, he might convert the published (Y, x, y) reflectances directly to *RGB* values using the inverse of M_{sRGB} given earlier. Unfortunately, he makes at least three mistakes in doing so. First, he is forgetting to perform a white point transform, so there is a slight red shift as he converts from (Y, x, y) under the bluish illuminant *C* to the more neutral D65 white point of *sRGB*. Second, the tungsten source in his model has a slight orange hue he forgets to account for, and there should be a general darkening of the surface under this illuminant, which he fails to simulate. Finally, the weak output at the blue end of a tungsten spectrum makes purple very difficult to distinguish from blue, and he has failed to simulate this metameric effect in his rendering. In the end, the rendering shows something more like violet than the dark blue one would actually witness for this color in such a scene.

If the spectra of all the light sources are equivalent, we can precompute the correct result for the direct-diffuse component and replace the light sources with neutral (white)

emitters, inserting our spectrally prefiltered *RGB* values as the diffuse reflectances in each material. We need not worry about how many spectral samples we can afford, since we only have to perform the calculation once for each material in a preprocess. If we intend to render in our display color space, we may even perform the white balance transform ahead of time, saving ourselves the final 3×3 matrix transform at each pixel.

In Section 3, we analyze the error associated with three different color spaces using our spectral prefiltering method, and compare it statistically to the error from naive rendering. The first color space we apply is CIE *XYZ* space, as recommended by Borges¹. The second color space we use is linear *sRGB*, which has the CCIR-709 *RGB* color primaries that correspond to nominal CRT display phosphors¹⁵. The third color space is the same one we apply in our white point transformation, the Sharp *RGB* space. We look at cases of direct lighting under a single illuminant, where we expect our technique to perform well, and mixed illuminants with indirect diffuse and specular reflections, where we expect prefiltering to work less effectively.

When we render in CIE *XYZ* space, it makes the most sense to go directly from the prefiltered result of Eq. (1) to *XYZ* colors divided by white under the same illuminant:

$$X'_m = \frac{X_m}{X_w} \quad Y'_m = \frac{Y_m}{Y_w} \quad Z'_m = \frac{Z_m}{Z_w}$$

We may then render with light sources using their absolute *XYZ* emissions, and the resulting *XYZ* direct diffuse component will be correct in absolute terms, since they will be remultiplied by the source colors. The final white point adjustment may then be combined with the display color transform exactly as shown in Eq. (2).

When we render in *sRGB* space, it is more convenient to perform white balancing ahead of time, applying both Eq. (1) and Eq. (2) prior to rendering. All light sources that match the spectrum of the dominant illuminant will be modeled as neutral, and spectrally distinct light sources will be modeled as having their *sRGB* color divided by that of the dominant illuminant.

When we render in the Sharp *RGB* space, we can eliminate the transformation into another color space by applying just the right half of Eq. (2) to the surface colors calculated by Eq. (1):

$$\begin{bmatrix} R'_m \\ G'_m \\ B'_m \end{bmatrix} = \begin{bmatrix} \frac{1}{R_w} & 0 & 0 \\ 0 & \frac{1}{G_w} & 0 \\ 0 & 0 & \frac{1}{B_w} \end{bmatrix} M_{\text{Sharp}} \begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix},$$

Dominant illuminants will again be modeled as neutral, and spectrally distinct illuminants will use:

$$R'_s = \frac{R_s}{R_w} \quad G'_s = \frac{G_s}{G_w} \quad B'_s = \frac{B_s}{B_w}$$

The final transformation to the display space will apply the

remaining part of Eq. (2):

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = M_D M_{\text{Sharp}}^{-1} \begin{bmatrix} R'_w & 0 & 0 \\ 0 & G'_w & 0 \\ 0 & 0 & B'_w \end{bmatrix} \begin{bmatrix} R'_m \\ G'_m \\ B'_m \end{bmatrix}.$$

3. Results

Our test scene was constructed using published spectral data and simple geometry. It consists of a square room with two light sources and two spheres. One sphere is made of a smooth plastic with a 5% specular component, and the other sphere is made of pure, polished gold (24 carat). The diffuse color of the plastic ball is Macbeth Green⁹. The color of elemental gold is computed from its complex index of refraction as a function of wavelength. The ceiling, floor, and far wall are made of the Macbeth Neutral.8 material. The left wall is Macbeth Red, and the right wall is Macbeth Blue. The near wall, seen in the reflection of the spheres, is the Macbeth BlueFlower color. The left light source is a 2856°K tungsten source (i.e., Standard Illuminant A). The right light source is a cool white fluorescent.

All spectral data for our scene were taken from the material tables in Appendix G of Glassner’s *Principles of Digital Image Synthesis*⁵, and these are also available in the Materials and Geometry Format (MGF)¹⁸. For convenience, the model used in this paper have been prepared as MGF files and included with our image comparisons in the supplemental materials.

Figure 2 shows a Monte Carlo path tracing of this environment with fluorescent lighting using 69 evenly spaced spectral samples from 380 to 720 nm, which is the resolution of our input data. Using our spectral prefiltering method with the cool white illuminant, we recomputed the image using only three *sRGB* components, taking care to retrace exactly the same ray paths. The result shown in Figure 3 is nearly indistinguishable from the original, with the possible exception of the reflection of the blue wall in the gold sphere. This can be seen graphically in Figure 5, which plots the CIE 1994 Lab ΔE^* color difference¹⁰ in false color. A ΔE^* value of one is just noticeable if the colors are adjacent, and we have found values above five or so to be visible in side-by-side image comparisons.

Using a naive assumption of an equal-energy illuminant, we recomputed the *sRGB* material colors from their reflectance spectra and rendered the scene again, arriving at Figure 4. The rendering took the same time to finish, about a third as long as the full-spectral rendering, and the results are quite different. Both the red wall and the green sphere have changed lightness and saturation from the reference image, the blue wall is reflected as purple in the gold sphere, and the ΔE^* errors shown in Figure 6 are over 20 in large regions. Clearly, this level of accuracy is unacceptable for critical color evaluations, such as selecting a color to repaint the living room.

Illum	Method	XYZ		<i>sRGB</i>		Sharp	
		50%	98%	50%	98%	50%	98%
tung	naive	10.4	45.9	4.8	15.4	0.8	5.1
	prefilt	2.3	5.7	0.6	1.5	0.5	0.9
fluor	naive	6.1	32.0	5.8	39.2	1.1	6.0
	prefilt	2.0	6.6	0.4	1.2	0.4	0.8
both	naive	5.6	31.6	4.5	21.5	0.6	2.8
	prefilt tung	4.9	15.1	0.5	2.0	0.7	2.2
	prefilt fluor	4.8	55.1	0.6	6.5	0.7	8.6
	Average	5.7	27.4	2.8	12.5	0.7	3.8

Table 1: CIE 1994 Lab ΔE^* percentiles for our example scene.

We repeated the same comparisons in CIE XYZ and Sharp RGB color spaces, then changed the lighting configuration and ran them again. Besides the fluorescent-only lighting condition, we looked at tungsten-only and both sources together. Since the lumen output of the two sources is equal, it was not clear which one to choose as the dominant illuminant, so we applied our prefiltering technique first to one source then to the other. Altogether, we compared 21 combinations of light sources, color spaces, and rendering methods to our multispectral reference solution. The false color images showing the ΔE^* for each comparison are included in the supplemental materials, and we summarize the results statistically in Table 1 and Figure 7.

Table 1 gives the 50th percentile (median) and 98th percentile ΔE^* statistics for each combination of method, lighting, and color space. These columns are averaged to show the relative performance of the three rendering color spaces at the bottom. Figure 7 plots the errors in Table 1 as a bar chart. The 50th percentile errors are coupled with the 98th percentile errors in each bar. In all but one simulation, the Sharp RGB color space keeps the median error below the detectable threshold, and the majority of the Sharp renderings have 98pixels below a ΔE^* of five relative to the reference solution, a level at which it is difficult to tell the images apart in side-by-side comparisons. The smallest errors are associated with the Sharp color space and spectral prefiltering with a single illuminant, where 98% of the pixels have errors below the detectable threshold. In the mixed illuminant condition, spectral prefiltering using tungsten as the dominant illuminant performs slightly better than a naive assumption, and prefiltering using cool white as the dominant illuminant performs slightly worse. The worst performance by far is seen when we use CIE XYZ as the rendering space, which produces noticeable differences above five for over 2% of the pixels in every simulation, and a median ΔE^* over five in each naive simulation.

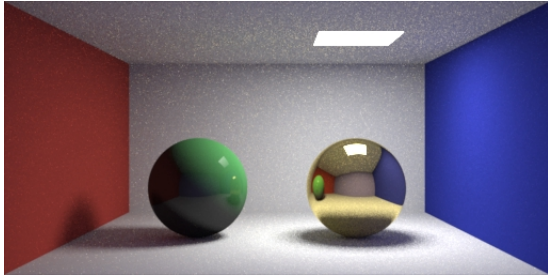


Figure 2: Our reference multi-spectral solution for the fluorescent-only scene.

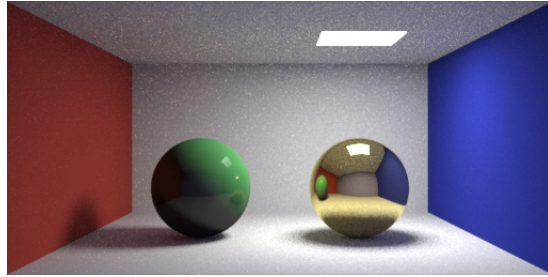


Figure 3: Our prefiltered sRGB solution for the fluorescent-only scene.

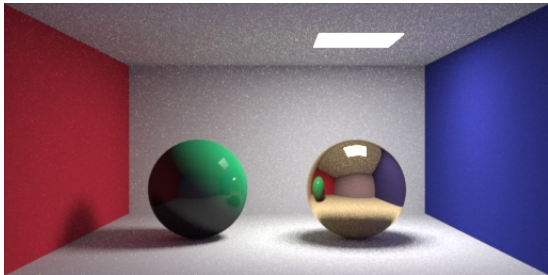


Figure 4: Our naive sRGB solution for the fluorescent-only scene.

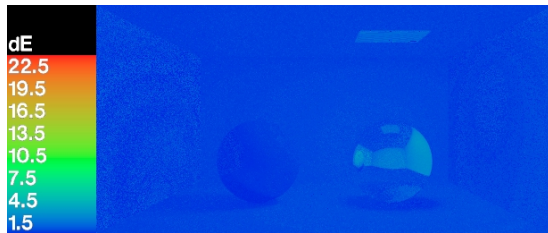


Figure 5: The ΔE^* error for the prefiltered sRGB solution.

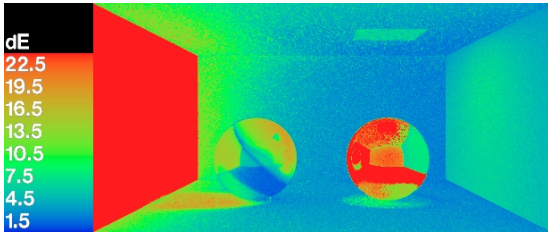


Figure 6: The ΔE^* error for the naive sRGB solution.

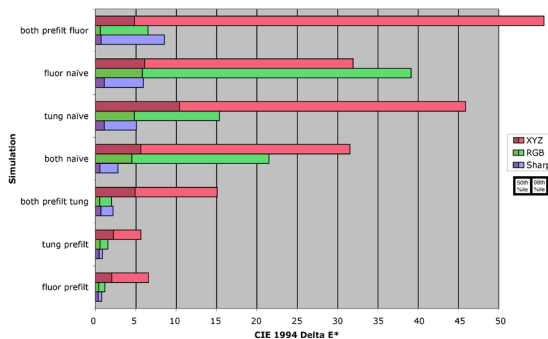


Figure 7: Error statistics for all solutions and color spaces.

4. Conclusions

In our experiments, we found spectral prefiltering to minimize color errors in scenes with a single dominant illuminant spectrum, regardless of the rendering color space. The median CIE 1994 Lab ΔE^* values were reduced by a factor of six on average, to levels that were below the detectable threshold when using the sRGB and Sharp color spaces. Of the three color spaces we used for rendering, the CIE XYZ performed the worst, generating median errors that were above the detectable threshold even with prefiltering, and five times the threshold without prefiltering, meaning the difference was clearly visible over most of the image in side-by-side comparisons to the reference solution. In contrast, the Sharp RGB color space, favored by the color science

community for chromatic adaptation transforms, performed exceptionally well in a rendering context, producing median error levels that were at or below the detectable threshold both with and without prefiltering.

We believe the Sharp RGB space works especially well for rendering because it minimizes the representation error for tristimulus values because its axes are aligned along the densest regions of XYZ space, perceptually. This property is held in common with the AC_1C_2 color space recommended by Meyer for rendering for this reason¹¹. In fact, the AC_1C_2 space has also been favored for chromatic adaptation, indicating the strong connection between rendering calculations and von Kries style transforms. This is evident in the diagonal matrix of Eq. (2), where white point primaries are mul-

tiplied in separate channels, analogous to the color calculations inside a three-component shader. Just as a white point shifts in a von Kries calculation, so do colors shift as they are reflected by a material.

The combination of spectral prefiltering and the Sharp *RGB* space is particularly effective. With prefiltering under a single illuminant, 98% of the pixels were below the detectable error threshold using the Sharp *RGB* space, and only a single highlight in the gold sphere was distinguishable in our side-by-side comparisons. We included a polished gold sphere because we knew its strong spectral selectivity and specularly violated one of our key assumptions, which is that the direct-diffuse component dominates the rendering. We saw in our results that the errors using prefiltering for the gold sphere are no worse than without, and it probably does not matter whether we apply our prefiltering method to specular colors or not, since specular materials tend to reflect other surfaces more than light sources in the final image, anyway. However, rendering in a sharpened *RGB* space always seems to help.

We also tested the performance of prefiltering when we violated our second assumption of a single, dominant illuminant spectrum. When both sources were present and equally bright, the median error was still below the visible threshold using prefiltering in either the *sRGB* or Sharp color space. Without prefiltering, the median jumped significantly for the *sRGB* space, but was still below threshold for Sharp *RGB* rendering. Thus, prefiltering performed no worse on average than the naive approach for mixed illuminants, which was our goal as stated in the introduction.

In conclusion, we have presented an approach to *RGB* rendering that works within any standard framework, adding virtually nothing to the computation time while reducing color difference errors to below the detectable threshold in typical environments. The spectral prefiltering technique accommodates sharp peaks and valleys in the source and reflectance spectra, and user-selection of a dominant illuminant avoids most white balance problems in the output. Rendering in a sharpened *RGB* space also greatly improves color accuracy, independent of prefiltering. Work still needs to be done in the areas of mixed illuminants and colored specular reflections, and we would like to test our method on a greater variety of example scenes.

Acknowledgments

The authors would like to thank Maryann Simmons for providing timely reviews of the paper in progress, and Albert Meltzer for critical editing and LaTeX formatting assistance. We also wish to thank the anonymous reviewers, who we hope will make themselves anonymous at the workshop so we may discuss their points in person, as there was not room to include such discussion in a short paper.

References

1. C. Borges. Trichromatic Approximation for Computer Graphics Illumination Models. *Proc. Siggraph '91*.
2. Anthony J. Calabria and Mark D. Fairchild. Herding CATs: A Comparison of Linear Chromatic-Adaptation Transforms for CIECAM97s. *Proc. 9th Color Imaging Conf.*, pp. 174–178, 2001.
3. Kristin J. Dana, Bram van Ginneken, Shree K. Nayar and Jan J. Koenderink. Reflectance and Texture of Real World Surfaces. *ACM TOG*, **15**(1):1–34, 1999.
4. G. D. Finlayson and P. Morovic. Is the Sharp Adaptation Transform more plausible than CMCCAT2000? *Proc. 9th Color Imaging Conf.*, pp. 310–315, 2001.
5. Andrew S. Glassner. Principles of Digital Image Synthesis. Morgan Kaufmann, 1995.
6. G. W. Larson, H. Rushmeier and C. Piatko. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Transactions on Visualization and Computer Graphics*, **3**(4) (December 1997).
7. Laurence T. Maloney. Evaluation of Linear Models of Surface Spectral Reflectance with Small Numbers of Parameters. *J. Optical Society of America A*, **3**(10):1673–1683 (October 1986).
8. David Marimont and Brian Wandell. Linear Models of Surface and Illuminant Spectra. *J. Optical Society of America A*, **9**(11):1905–1913 (November 1992).
9. C. S. McCamy, H. Marcus and J. G. Davidson. A color-rendition chart. *J. Applied Photographic Engineering*, **2**(3):95–99 (summer 1976).
10. R. McDonald and K. J. Smith. CIE94 - a new color-difference formula. *Soc. Dyers Col.*, **111**:376-9, Dec 1995.
11. Gary Meyer. Wavelength Selection for Synthetic Image Generation. *Computer Vision, Graphics and Image Processing*, **41**:57–79, 1988.
12. Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild and Donald P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. *Proc. Siggraph '98*.
13. Mark S. Peercy. Linear color representations for full speed spectral rendering. *Proc. Siggraph '93*.
14. Brian Smits. An RGB to Spectrum Conversion for Reflectances. *J. Graphics Tools*, **4**(4):11–22, 1999.
15. Michael Stokes et al. A Standard Default Color Space for the Internet – sRGB. Ver. 1.10, November 1996. <http://www.w3.org/Graphics/Color/sRGB>.
16. S. Sueeprasan and R. Luo. Incomplete Chromatic Adaptation under Mixed Illuminations. *Proc. 9th Color Imaging Conf.*, pp. 316–320, 2001.

17. S. Süsstrunk, J. Holm and G. D. Finlayson. Chromatic Adaptation Performance of Different *RGB* Sensors. *IS&T/SPIE Electronic Imaging*, SPIE **4300**, Jan. 2001.
18. Greg Ward et al. Materials and Geometry Format. <http://radsite.lbl.gov/mgf>.
19. Harold B. Westlund and Gary W. Meyer. A BRDF Database Employing the Beard-Maxwell Reflection Model. *Graphics Interface 2002*.
20. Günter Wyszecki and W. S. Stiles. Color Science: Concepts and Methods, Quantitative Data and Formulae. John Wiley & Sons, New York, 2nd ed., 1982.