Color Enhancement and Rendering in Film and Game Production: Film Simulation for Video Games

Yoshiharu Gotanda
tri-Ace, Inc.

1. Introduction

In video games, High Dynamic Range (HDR) rendering is commonly used and allows us to render more photo-realistic looking images. HDR rendering uses a tone-mapping process to achieve a Low Dynamic Range (LDR) rendering result for a display. The process compresses or clamps HDR color information to fit LDR devices which typically have an 8-bit color resolution. There are a lot of tone-mapping algorithms, however, typical ones are designed for compressing HDR color information efficiently and may leave the viewer artistically unsatisfied. On the other hand, in photography, film is used to record an image from the real world, which can be thought of as tone-mapping from a computer graphics point of view. Therefore, we think of film as a standard for tone-mapping because film manufacturing engineers have been trying to improve the quality of film to reproduce high fidelity or memory colors[1].

In order to use film characteristics for tone-mapping, we referenced the following figures[1] from film specification documents: Spectral Sensitivity Curves, H-D Curves and Spectral Dye-density Curves. Figure 1 shows those example curves. According to the documentation from KODAK[2], Spectral Sensitivity Curves describe the relative sensitivity of the emulsion. The H-D Curves plot the amount of exposure against the density achieved by that exposure. Spectral Dye-density Curves indicate the total absorption by each color dye measured at a particular wavelength of light and the visual neutral density (at 1.0) of the combined layers measured at the same wavelength. H-D Curves are also called Hurter-Driffield Curves, Characteristic Curves, D-logE Curves or D-logH Curves.

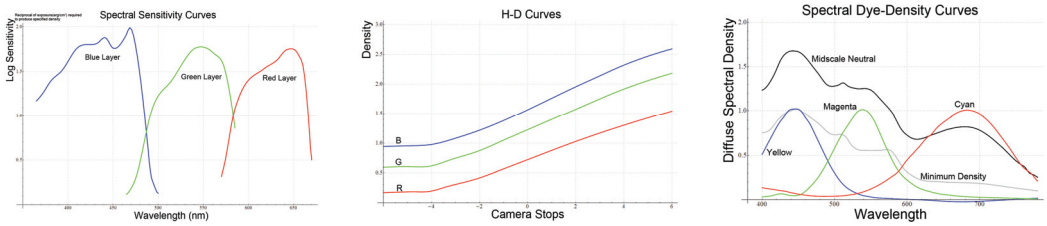

Figure 1: *The left graph shows Spectral Sensitivity Curves, the middle shows H-D Curves and the right shows Spectral Dye-density Curves.*

2. First Implementation

For the first implementation, we used these curves to get appropriate parameters for our previous tone-mapping pipeline. Our first tone-mapping pipeline used two matrix operations and one texture fetch. The texture is pre-computed on the CPU with tone mapping operators and stores LDR information. In the shader, input HDR information is multiplied with a pre-color matrix to get a temporary color vector. Then, the

[1] Memory color is color memorized as an image in the brain. When talking about psychological effects for colors, this word is often used.

temporary vector is used for three 1D texture fetches to get R, G, and B components. Lastly, we multiply the R, G, and B components by a post-color matrix:

$$v'_{hdr} = M_{pre} \cdot v_{hdr},$$

$$v^R_{ldr} = f_R(v'^R_{hdr}), v^G_{ldr} = f_G(v'^G_{hdr}), v^B_{ldr} = f_B(v'^B_{hdr}),$$

$$v_{out} = M_{post} \cdot v_{ldr}, \quad (1)$$

where v_{hdr} is the original color vector in the render target, M_{pre} and M_{post} are the pre- and post-color matrix, v_{out} is the result color vector, and functions of f are 1D LUT texture fetch used for non-linear color conversion.

The pre-color matrix is created with Spectral Sensitivity Curves. We sampled each R, G and B curve at 464, 549 and 612nm to create the nine values used to make the matrix. If there is no curve at the sample points, we extrapolate it. This can be thought of as a simplification of integrating each R, G, and B of Spectral Sensitivity Curves over the visible spectral domain. We tried a more accurate integral, but it made a subtle difference. As a result, we did not use it.

R, G and B of H-D Curves are used for creating the 1D lookup texture with gamma correction. Spectral-Dye-density Curves are not used for this implementation. Therefore, the post-color matrix is only used for manual color filters. When negative films are chosen, a duplication process is necessary to render colors correctly. We also apply the print film specification to the texture. However, due to a limitation of our shader implementation, print film's Spectral Sensitivity Curves are not applied. Figure 2 shows the results of the different film parameters.

3. Film simulation

The first approach could not be called a "simulation." Film data was only used for determining default parameters for the existing tone-mapping pipeline. Moreover, no film specific characteristics were reproduced at all because only the color matrix and non-linear tone curves were applied in the tone-mapping pipeline, as a filmic tone-mapping.

Instead of using the existing tone-mapping pipeline, we implemented the new tone-mapping pipeline from scratch to reproduce film specific characteristics. To that end, we try to simulate each color process in film as best as possible, referring to film specification sheets. However, because we cannot get enough information from the sheets, we interpolate or complement the missing information to simulate the processes appropriately.

Figure 2: *Comparison with different film settings and Reinhard. From the top left, Reinhard, K-Reversal, F-Reversal, F-Reversal 2 and K-Negative.*

In the beginning, our rendering pipeline didn't use any physical unit for the render target. However, because Spectral Sensitivity Curves require spectral irradiance with physical units such as erg/cm^2, we use the physical unit "W/m^2" (watt per square meter^2) for the parameter of light intensity. Including an area in the unit is convenient[3] because otherwise it would have to be calculated in the shader. Ideally, we should use radiance or radiant flux for a light source instead of irradiance. However, since our entire rendering pipeline is a coarse approximation of the rendering equation, in some places the calculation of the area is ignored for performance.

As for helper functions, we implemented some unit conversion ones such as lx (lux) or lm (lumen) from/to W/m^2 and color temperature from/to RGB. After using a physical unit for light sources, we also apply it to the render target. When using a low precision frame buffer such as an 8bit or 10bit format, we convert the physical unit in the render target to mJ / m^2 / F / ISO100 (100・mJ・m^{-2}・F^{-1}・ISO^{-1}). If we can use a high precision frame buffer, the unit is converted in the tone-mapping pass. The scale factor c for the conversion is decided by:

---

[2] "W/m^2" indicates irradiance.

[3] Our engine implements a (D)SLR camera based simulation, that requires a film (sensor) size and that each pixel size is defined. If the physical light amount in the rendering pipeline doesn't have area, we always have to compute the area properly. As a simple example, if each pixel has light energy of 1J instead of 1J/m^2, you have to compute how much light energy is received considering the area (or radiance). Then, in the film simulation pass, since film specification sheets require "erg/cm^2", it's again converted to 1J/m^2 from 1J.

$$c = \frac{1000 \cdot t}{F \cdot \frac{ISO}{100}}, \tag{2}$$

where t is the time value that is typically equivalent to shutter speed[4], F is the F stop[5] of the camera and ISO is the ISO sensitivity[6] set in the camera. "1,000" in the numerator is for unit conversion from Watt to Milliwatt.

After acquiring a physical amount of light energy in the render target, we reconstruct spectral information from the buffer:

$$I_\lambda = M \cdot v_{r,g,b}. \tag{3}$$

Output I_λ is a vector that contains discrete spectral data. M is the reconstruction matrix and $v_{r,g,b}$ is a physical RGB color vector in the render target. In our implementation, we treat the spectral vector at 5nm intervals from 380nm to 780nm. However, since the RGB color vector only has 3 numbers, the spectral vector can be reconstructed using any method so long as the spectral curves are continuous. Therefore we restrict the spectrum reconstruction with the following rules.

We use two matrices to define the rules and to perform the conversion. The first matrix M_a is used for spectrum reconstruction from an RGB color vector and the second matrix M_b is used for conversion from spectral intensity to an RGB color vector. The second matrix is designed uniquely based on Color Matching Function Table[7] in white papers about XYZ color space[8]. Therefore, we should determine restrictions for creating the first matrix. Table 3 in the Appendix shows our second matrix.

When an input RGB color vector is multiplied with M_a and the resulting spectral vector is converted via multiplying by M_b to create an output RGB color vector there are restrictions for both vectors which are shown in Table 1. In addition to these restrictions, we tried to keep spectral curves' shape as continuous as possible. We didn't use any numerical methods[9] to create the matrix and tuned it by hand.

Input color vector	*Restriction*
(1, 1, 1)	Output vector should be (1, 1, 1)
(1, 0, 0)	Output vector should be (1, 0, 0)
(0, 1, 0)	Output vector should be (0, 1, 0)
(0, 0, 1)	Output vector should be (0, 0, 1)
Any	Magnitude of both input and output vectors should be same

Table 1: *Design restrictions for the spectrum reconstruction matrix.*

[4] Shutter speed is a unit for a camera. It means that the length of time a camera's shutter opens during exposure.

[5] F stop is a unit for the camera aperture. It means the focal length divided by the effective aperture diameter. The number doubles, the amount of light through the aperture decreases by fourfold.

[6] ISO sensitivity is a unit of film speed. The sensitivity of a film is in proportion to the number of ISO. In our implementation, ISO 100 is used for the standard value.

[7] If only RGB data is used as input energy, both matrices can be designed in any way as long as color vector values are preserved. However, if we do spectral rendering in the near future, the result is already a spectral value and doesn't need to be multiplied by the M_a before being passed to the film simulation. However, M_b still needs to convert this spectral data to sRGB so it can be displayed on a TV monitor. A Color Matching Function Table can be one of the best solutions for converting spectrum data to a sRGB color vector.

[8] This table can be found on a lot of papers, books and websites.

[9] If we only follow the explicit restrictions, a numerical method such as a non-linear least square would easily compute the matrix. However, we implicitly try to keep peak of RGB close to sRGB color primaries or to control the shape of spectra. (If you widen each spectrum, it will have better continuity and cause less saturation. If you keep each spectrum sharper, it will have discontinuity and better saturation.) We would like to control these nuances by hand. Anyway, our matrix can be still improved.

Figure 3 shows the difference of rendering results with different reconstruction matrices and examples of spectra with some RGB color vectors.

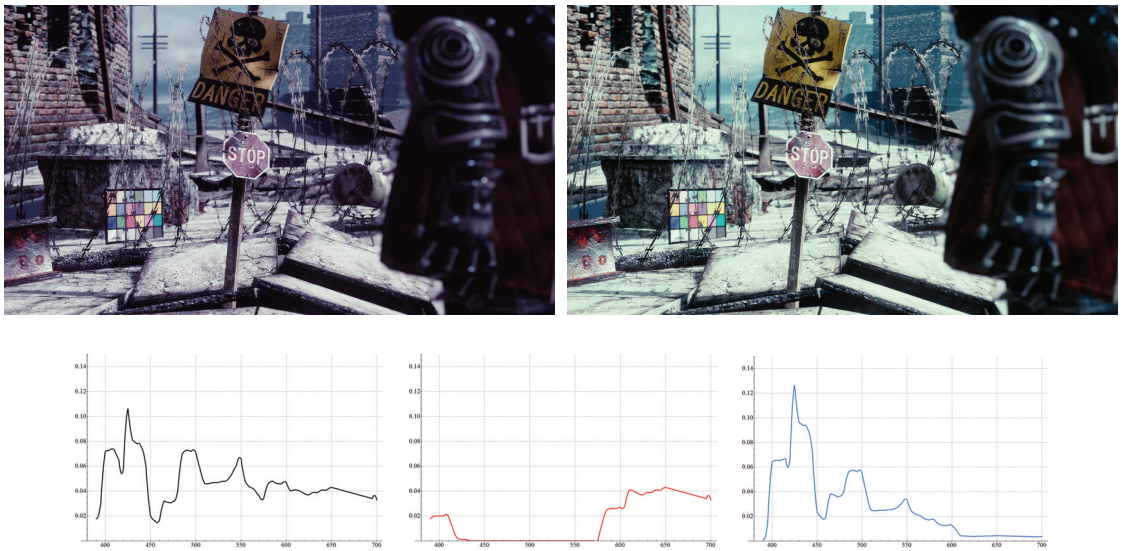

Figure 3: *Images use different matrices. The image on the left uses our standard matrix and the image on the right uses a peaky matrix. However, since the color balance of peaky matrix is not adjusted properly, it is just for a reference. The graphs are spectra reconstructed by our standard matrix. Graphs from the left, reconstructed from color vector (1, 1, 1), (1, 0, 0) and (0.1, 0.5, 1.2). This can be obviously improved and we are still researching to do so.*

After acquiring the spectral input vector, we do virtual exposure using Spectral Sensitivity Curves and H-D Curves and acquire the densities of three dyes. The following equation is used for virtual exposure:

$$D_{r,g,b} = f_{r,g,b}\left(\log_{10}\left((diag(c_{r,g,b}) \cdot w_{r,g,b}) \cdot I_{\lambda}\right)\right), \tag{4}$$

where $D_{r,g,b}$ is the result densities of three dyes, $f_{r,g,b}$ is the H-D Curves function, $c_{r,g,b}$ is the matching vectors, $w_{r,g,b}$ is the vector that is discretized from Spectral Sensitivity Curves converted to linear space and I_{λ} is the input spectral energy from Equation 3. This equation is evaluated three times for each R, G and B dye. The density is a unit that describes how much light gets through (transmission) and how much light doesn't (opacity, the reciprocal of transmittance). The density D can be represented by the following equation:

$$D = \log_{10}\frac{P_o}{P_t}, \tag{5}$$

where P_o is the light incident on processed film and P_t is the light transmitted by the film. The vectors $c_{r,g,b}$ in Equation 4 are used to connect Spectral Sensitivity Curves and H-D Curves. If we have perfect 3D Spectral Sensitivity Surfaces, they contain both Spectral Sensitivity Curves and H-D Curves. Therefore, we do not need the matching vectors. However, typical specification sheets only have 2D curves, therefore we have to compensate to make 3D surfaces.

We assume that Spectral Sensitivity Curves and H-D Curves together form 3D surfaces[10] as illustrated in Figure 4.

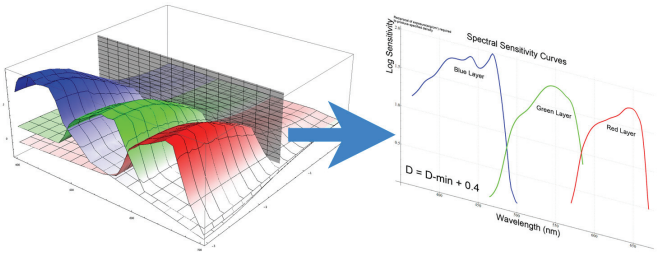

Figure 4: *The image of relationship between virtual 3D surfaces and the Spectral Sensitivity Curves. Spectral Sensitivity Curves can be thought of the curves sliced from the original 3D surfaces at the specified density, D. For each RGB dye, every reading of the Spectral Sensitivity Curves is assumed to take place at the same density on the H-D curves, as calculated by the specification sheet's provided Density formula.*

For Spectral Sensitivity Curves, "Log Sensitivity" in the vertical axis stands for "Reciprocal of absolute exposure (erg/cm^2) in logarithmic space required to produce the specified density". For H-D Curves, "Log Exposure" in the horizontal axis, "Relative exposure in logarithmic space."

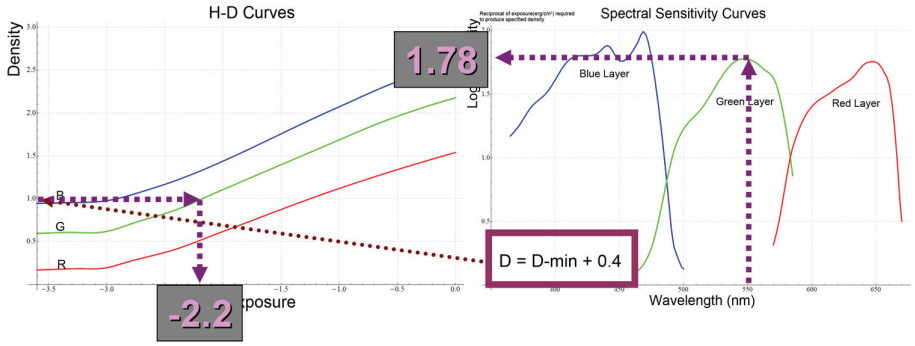

Figure 5: *The image explains how to find values to calculate a component of each matching vector. Note both values are in logarithmic space and the log sensitivity "1.78" is reciprocal of exposure. Calculate* $10^{-2.2}/10^{-1.78} \approx 0.38$ *which is a component of a matching vector for the green dye at 550nm.*
Note that since a film specification sheet defines D, it, and therefore the numerator will be the same for every nm sampling of a single Spectral Sensitivity Curves graph.

In order to convert "relative Log Exposure" in H-D curves to "absolute Log exposure", we need the matching vectors. For each RGB dye, we determine density D and based on H-D Curves arrive at the required log

[10] Usually, specification sheets will contain a single Spectral Sensitivity Curves graph and a single H-D Curves graph. Additionally, the specification sheet will give some definition for density D such as "D = D-min + 0.4" where D-min is the minimum density or "D = 1.0", though the former is more common. For example, looking at Green layer of the Spectral Sensitivity Graph as shown on Figure 5, at 550nm, a Log Sensitivity value of 1.78 exists. The graph defines Log Sensitivity as "Reciprocal of exposure (erg/cm^2) required to produce the specified density". Thus, the graph can be read as follows: $10^{1/1.78}$ is amount of light needed at 550nm to produce a density as defined by the specification sheet. A second example reading at 500nm produces a Log sensitive value of 1.25 which means that $10^{1/1.25}$ amount of light is needed at 500nm to produce a density as defined by the specification sheet.

exposure. This value gives us the numerator for our matching vectors. To calculate the denominators, we use Spectral Sensitivity Curves and chose a range of wavelengths to determine log sensitivity values for each RGB component. One such calculation is illustrated by Figure 5 above.

Now we have the densities of each dye. However, if you use a negative camera film, it must be duplicated to print film. Negative film cannot be seen with proper colors using a projector. Duplication is a simple process involving shooting a negative film with print film. Intermediate film and/or a digital intermediate are also used in the film industry. However, for our simulation, they are not necessary.

In order to simulate the duplication process, a virtual light source is necessary for projection. We used a spectrum of tungsten light as shown in Figure 6. When only using this spectrum, the color for each dye is not separated enough. Typically, the documentation of print film describes how to duplicate film to print film and filters for color separation. Figure 6 also shows color filters that we used and Table 2 shows Neutral Density (ND) filters that we used.

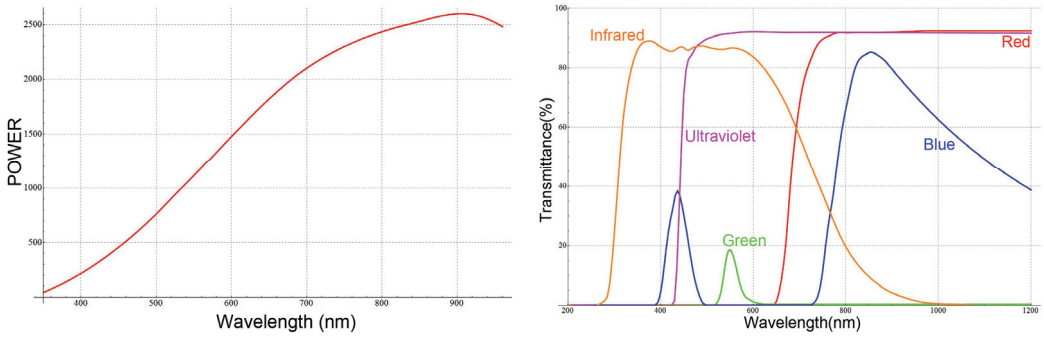

Figure 6: *The graph on the left shows the spectrum of tungsten light used for virtual projection. The graph on the right shows color filters. Red, Blue and Green Lines are each color's pass filter. The Yellow line is an infrared cut filter and the Pink line is ultraviolet cut filter. The domain over 700nm of the blue color filter should be removed because it produces undesired color. The dichroic filter in an additive printing system may not have that domain.*

	Scale
Red	0.5
Green	0.55
Blue	0.9

Table 2: *Neutral Density filters used for virtual projection*

Using these color filters, ND filters and tungsten light, the spectrum of projection light can be calculated with:

$$l(\lambda) = t(\lambda) f_{UV}(\lambda) f_{IR}(\lambda)\left(n_R f_R(\lambda) + n_G f_G(\lambda) + n_B f_B(\lambda)\right), \tag{6}$$

where $t(\lambda)$ is the spectrum of the tungsten light, $f_{UV}(\lambda)$ is the spectrum of the ultraviolet filter, $f_{IR}(\lambda)$ is the spectrum of the infrared filter, $n_{R,G,B}(\lambda)$ is the scale of each Neutral Density filter and $f_{R,G,B}(\lambda)$ is the spectrum of each color filter.

Using Spectral Dye-density Curves and H-D Curves with the densities from Equation 4, virtual projection can be rendered with the following equation:

$$\sigma(\lambda) = l(\lambda) \cdot 10^{-\left(S_{\min}(\lambda) + c \cdot \sum\limits_{r,g,b} \left(S_{r,g,b}(\lambda) \frac{D_{r,g,b} - D_{r,g,b}^{\min}}{D_{r,g,b}^{mid} - D_{r,g,b}^{\min}} \right) \right)}, \tag{7}$$

where $l(\lambda)$ is the spectrum of the projection light from Equation 6, $S_{\min}(\lambda)$ is the minimum density spectrum shown in Spectral Dye-density Curves of the negative film, c is the matching constant as explained later, $S_{r,g,b}(\lambda)$ is each Spectral Dye-density Curve of the negative film, $D_{r,g,b}$ is the densities of the negative film computed by Equation 4, $D_{r,g,b}^{\min}$ is the minimum densities[11] of each dye found in H-D Curves, and $D_{r,g,b}^{mid}$ is the midscale densities used for the target density. Output spectrum $\sigma(\lambda)$ is used as the input spectrum vector for Equation 4 for computing the densities of the exposed print film.

$D_{r,g,b}^{\min}$ is the minimum value of each H-D Curves. $D_{r,g,b}^{mid}$ is the midscale densities which are typically located at camera stop of 0 on the H-D Curves. If the H-D Curves do not have camera stop on the horizontal axis, then the midscale densities can be chosen from the point that indicates an 18% gray with the proper exposure.

As mentioned before about 3D Spectral Sensitivity Surfaces, Spectral Dye-density Curves should be also 3D surfaces. However, from typical specification sheets, only 2D curves for a certain density can be obtained. Therefore, once again we assume that Spectral Dye-density Curves are scaled with the density. The issue is whether the curves should be scaled in linear space or logarithmic space. First, we assume that the midscale density (or visual neutral density) can be approximated with the following equation:

$$S_{mid}(\lambda) \approx S_{\min}(\lambda) + c \cdot \sum_{r,g,b} S_{r,g,b}(\lambda), \tag{8}$$

where $S_{mid}(\lambda)$ is the midscale density[12] shown in the Spectral Dye-density Curves, and the other functions and coefficients are shown in Equation 7. In order to obtain the coefficient (the matching constant c), we solve the following equation numerically in linear and logarithmic spaces:

$$c = \int_{\lambda} \frac{S_{mid}(\lambda) - S_{\min}(\lambda)}{\sum\limits_{r,g,b} S_{r,g,b}(\lambda)} d\lambda. \tag{9}$$

Figure 7 shows comparisons between the actual midscale density curve and fitted curve in linear and logarithmic spaces.

As a result of our experiment, we conclude that Equation 7 approximates 3D Spectral Dye-density Surfaces well using 2D Spectral Dye-density Curves and H-D Curves.

---

[11] These minimum densities are different from the minimum density in the Spectral Dye-Density Curves.

[12] This midscale density is different from the midscale densities in Equation 7.

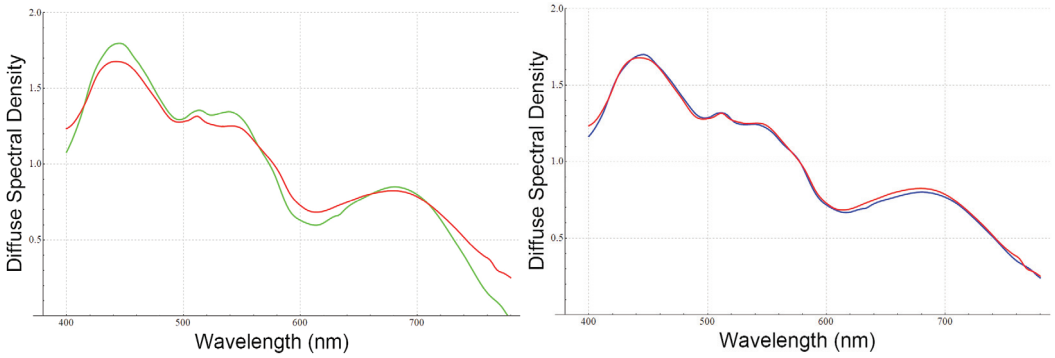

Figure 7: *The red lines in both graphs are the original midscale density curve. The green line is the result of the linear version[13] of Equation 8 with the fitted* c. *The blue line is the result of logarithmic version Equation 8. The graph on the right (scaling in logarithmic space) has much better correspondence than the graph on the left.*

Before the final calculation, we need to calibrate the spectrum of the projection light to acquire the correct white balance and brightness of the print film. We did it according to the print guide documentation. First, we assume that there is developed negative film shot[14] with an 18% gray card with the proper exposure. Using this developed film, the spectrum of the projection light is adjusted to get the specified densities according to the specification sheets. This computation is iteratively done in our implementation. However, because color separation is good enough with the color filters, the computation is not iterated in practice. For this calibration, we implemented two calibration modes. The first one is to calibrate the spectrum to the densities specified by print films' specification sheets. This method is described in the print film documentation. However, 18% gray doesn't become perfect gray due to developed negative film's Spectral Dye-density Curves balance. The second method is to calibrate the spectrum to get densities for Equivalent Neutral Density 0.7 (E.N.D. 0.7). With this method, regardless of the negative film's Spectral Dye-density Curves, neutral gray balance can be achieved. Figure 8 shows a comparison of two different calibration modes.

Figure 8: *Comparison of two calibration modes. The image on the left is calibrated based on the print film documentation. The image on the right is calibrated based on Equivalent Neutral Density. Since, at the neutral gray point, the right image is calibrated to the ideal gray balance and has a natural color balance. The left image remains the characteristics from Spectral Dye-Density Curves compare to the right image.*

---

13 The linear version of Equation 8 is much more complicated and does not produce worthwhile result. So, it is not shown.

14 This film could be obtained and it is called Laboratory Aim Density (LAD) control film. But we haven't.

During implementation of the duplication process, the spectral data was computed between 400nm and 1,000nm. After the experiment, we tried reducing the range to 400nm to 800nm for optimization. However, since the color balance of the result changed more than expected, we reverted back to the original range. The reason why the process needs infrared domain may be color separation. Print films are probably designed for duplicating camera film or intermediate film as faithful as possible. In order to get good color separation, the spectral sensitivity of the print film's red dye gets closer to infrared domain than camera film. As a result, the red dye tends to be influenced from infrared. Therefore spectrum values over 800nm are necessary for our implementation. Figure 9 shows a comparison of print film results with and without infrared.

Figure 9: *The image on the left is rendered with the infrared domain. The image on the right is rendered without the infrared domain. They only have a small difference, though the left image is a little more reddish than the right on, because the light in the infrared domain influences the red dye.*

Finally, we need to project the duplicated print film or positive film on the display. When the film is projected, the Spectral Dye-density Curves show how much light from the light source passes through film in the spectral domain. Again, these curves should be provided as 3D graphs. However, since there are not, we used previously mentioned assumed curves. In theaters, xenon lamps are typically used as the light source and have a color temperature of about 5,500K. For video games, since sRGB color space uses 6,500K white balance, we use 6,500K ideal black-body light. With these assumptions, the final spectrum is computed with the following equation:

$$\sigma(\lambda) = l_p(\lambda) \cdot 10^{-\left(\sum_{r,g,b} S_{r,g,b}(\lambda) \cdot D_{r,g,b} \right)}. \tag{10}$$

This equation is similar to Equation 6. However, since print film or positive camera film do not have a minimum density, the equation can be simplified. In this equation, $l_p(\lambda)$ is the spectrum of 6,500K ideal black-body light. The output spectral vector is finally converted to a color vector in sRGB color space[15]. This conversion uses the matrix M_b introduced in the spectrum reconstruction before and in Table 3 in the Appendix. After acquiring a color vector, gamma correction is necessary because the vector is in linear color space.

These computations are too expensive to implement in real-time. Therefore, we precompute the values on a CPU and store the result in a volume texture as 3D Look-Up Table. We ideally wanted to use a 64x64x64

[15] This means that RGB primaries for spectrum to sRGB conversion are based on sRGB color space.

texture. However, according to our research, due to cache efficiency, 32x32x32 was better to use even with added decompression requirements. The texture is compressed in log-space and ISO sensitivity is handled with a scalar value in order to keep the values in a proper range. We empirically use the following equation for decompression:

$$U = 0.534577 + 0.217563 \cdot \log_2(u + 0.191406)\,. \tag{11}$$

This equation is used for calculating U, V and W positions in log space from the color vector.

4. Limitations and future work

Our implementation of film simulation is based on insufficient specifications and information. In order to achieve more accurate results, we need more information from yet unpublished film specifications. If we had more detailed information about 3D surfaces, we could directly use it via a 3D Look-Up Table.

Currently, because we use an RGB vector for lights and shading to represent color, we have to reconstruct a spectrum. However, our reconstruction matrix or method can't be said as the best one. We have to keep on improving them. On the other hand, if we can construct the entire rendering pipeline using a color spectrum, we can achieve more accurate film simulation and more correct results from other parts of the rendering pipeline.

Regarding film grain, we currently use grain simulation based on ISO sensitivity. However, using this film simulation and grain curves contained in the specification sheets, we can add a grain simulation stage in order to achieve a more realistic grain simulation.

5. Conclusion

Using film specification sheets for color rendering instead of standard tone-mapping, we can get more complex results. During the research of 3D LUT compression, we tried to fit the texture to an equation. As a side effect, we found that the saturation of each color in the LUT is gradually changed according to the average of R, G and B values. Mathematically, it can be represented with affine color matrices including rotation and non-uniform scale factors based on the average of R, G and B. Filmic tone-curve and color balance variance can be achieved even with H-D Curves and 1D LUT. However, a more realistic filmic representation needs additional complex color variation on different color vectors. Lastly, comparisons using various film settings are shown in Figure 10.

Acknowledgements

The author would like to thank to Tatsuya Shoji for helping with the research and implementation of film simulation, Bart Sekura and Elliott Davis for reviewing the paper and slides, Kenichi Kanekura, Kazuki Shigeta, Kenichi Kaneko and Ryo Mizukami for creating beautiful graphics samples, and my fellow course speakers, especially Nathaniel Hoffman for reviewing.

References

[1] KODAK VISION 2 Color Negative Films. In *KODAK Web Site*.

[2] BASIC SENSITOMETRY AND CHARACTERISTICS OF FILM. In The Essential Reference Guide for Filmmakers on *Kodak Web Site.*

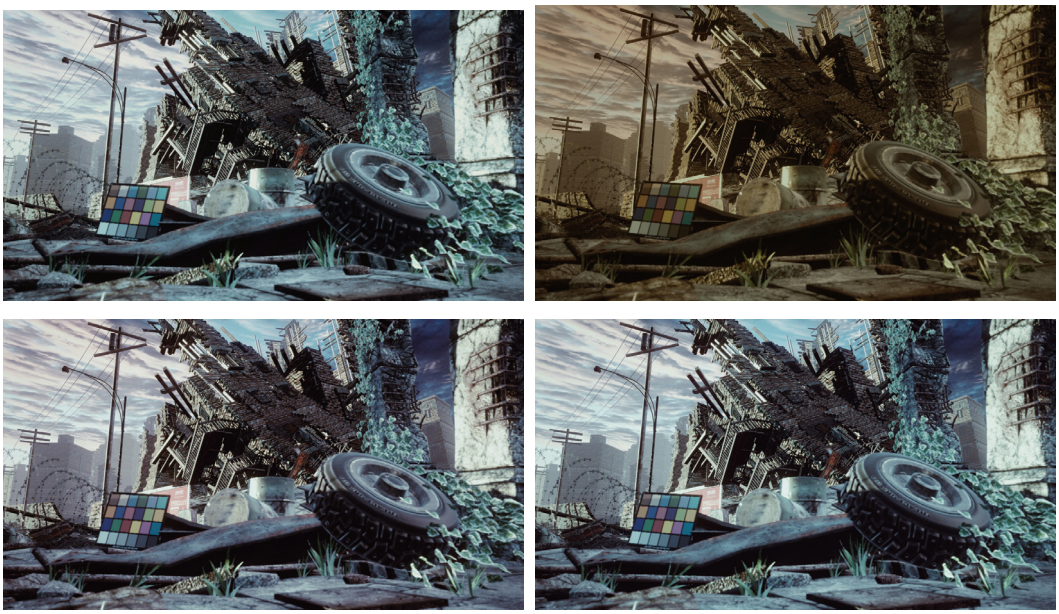

Figure 10 (a): *Print film comparison with our new film simulation. These images are rendered with the same camera negative film and different print films. Only the top right image is much different from the others, because the Spectral Density Curves of it are also different from the others'. The other print films look similar at a glance, however they are slightly different.*

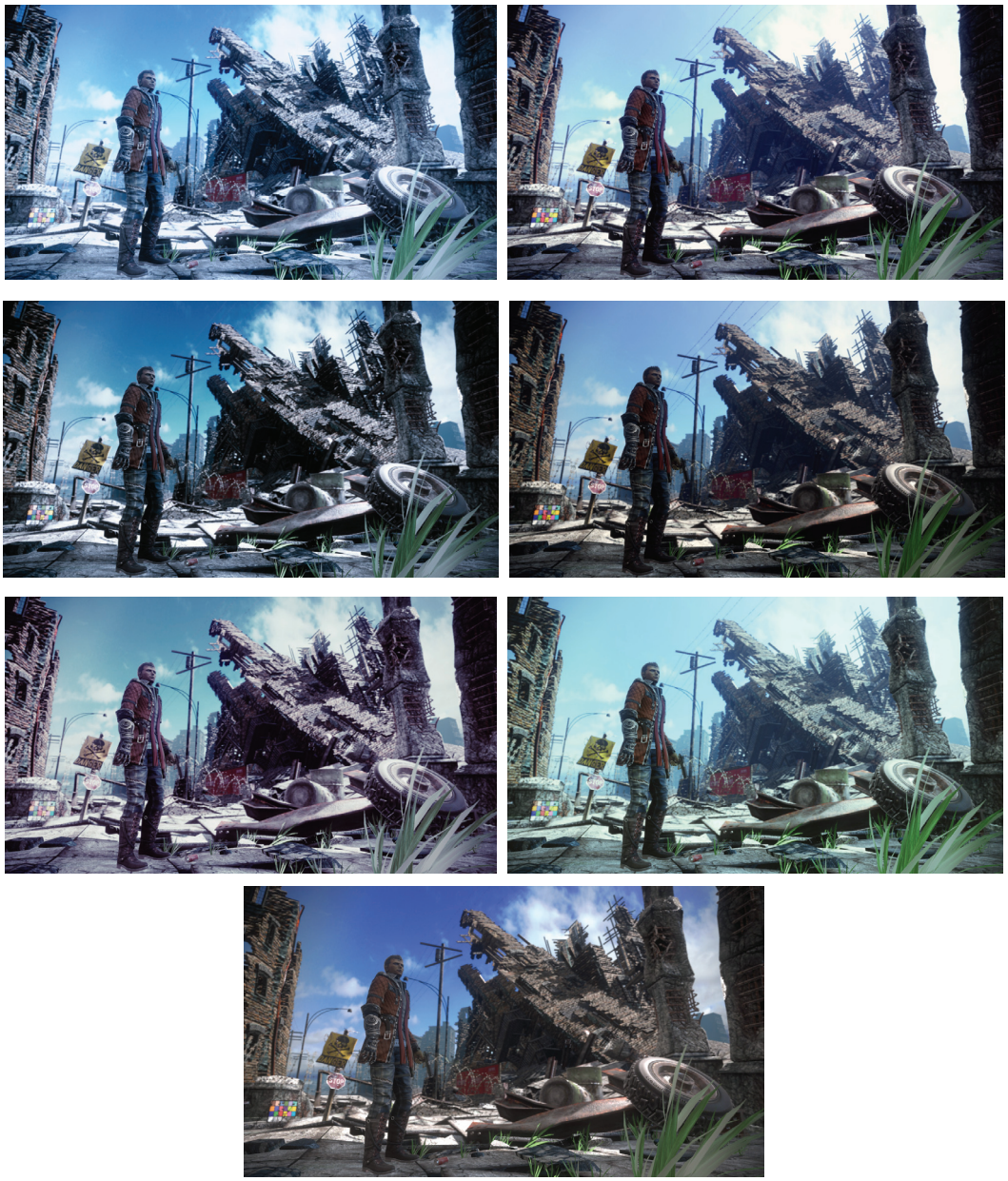

Figure 10 (b): *The left images are rendered with the new film simulation. The right images are rendered with the first implementation. The bottom image is rendered with Reinhard. From the top, K-Reversal, F-Reversal and K-Negative. Compared to the right images, the left images look to achieve filmic characteristics.*

Appendix

λ	R	G	B	λ	R	G	B
380	0.00115722	-0.000984624	0.006885892	585	1.91573676	0.5831081	-0.11053738
385	0.001887371	-0.001608584	0.011262839	590	2.16161892	0.42564418	-0.09610039
390	0.003568472	-0.003053872	0.021404705	595	2.35564314	0.27969829	-0.08184441
395	0.006402712	-0.005502321	0.038655807	600	2.47179324	0.15449742	-0.06871386
400	0.011934245	-0.010306367	0.072433733	605	2.51678724	0.0501465	-0.05675308
405	0.019219986	-0.016694979	0.117642523	610	2.475644436	-0.02787763	-0.0464078
410	0.035728854	-0.031280021	0.221398467	615	2.362646736	-0.08160284	-0.03748224
415	0.063086502	-0.055717513	0.396343371	620	2.183162736	-0.11318892	-0.029930305
420	0.107426868	-0.095905182	0.689068166	625	1.94149578	-0.12589551	-0.02352532
425	0.166691772	-0.151276738	1.108749339	630	1.67437851	-0.125332285	-0.01822547
430	0.21131466	-0.19580903	1.47802603	635	1.422493782	-0.117997065	-0.01405246
435	0.229442796	-0.219342338	1.73033081	640	1.182444768	-0.10570448	-0.01073083
440	0.222196452	-0.221802102	1.861349616	645	0.956762454	-0.090343145	-0.00808567
445	0.193310316	-0.207358594	1.897515942	650	0.7542297	-0.07397255	-0.00603705
450	0.147502074	-0.180921215	1.88409461	655	0.5832837	-0.05883315	-0.00446481
455	0.08938536	-0.14636988	1.85147329	660	0.44060574	-0.04534781	-0.00325907
460	0.01787136	-0.09993632	1.76830196	665	0.324232344	-0.033807516	-0.00234348
465	-0.06179508	-0.04125302	1.61411237	670	0.23403804	-0.02465626	-0.00165982
470	-0.148788144	0.03481304	1.353357112	675	0.17043912	-0.01810348	-0.00119028
475	-0.2320908	0.11677324	1.08623287	680	0.125430462	-0.013426853	-0.000862911
480	-0.30910743	0.201845545	0.836255218	685	0.088292316	-0.009517274	-0.00059915
485	-0.37969251	0.286997485	0.620014015	690	0.060941208	-0.006593712	-0.00041045
490	-0.447975486	0.378494397	0.451042137	695	0.042533708	-0.004612173	-0.000285204
495	-0.52603848	0.485501	0.32150249	700	0.030504381	-0.003311204	-0.000204112
500	-0.61625586	0.61242379	0.22188493	705	0.021782048	-0.00236453	-0.000145733
505	-0.7241769	0.77049843	0.14144558	710	0.015548789	-0.001687633	-0.000104061
510	-0.82195254	0.94108193	0.06512341	715	0.011034421	-0.001197523	-7.38647E-05
515	-0.8963172	1.11730212	-0.00438503	720	0.007785051	-0.000844879	-5.21137E-05
520	-0.925394688	1.273763072	-0.058605611	725	0.005502461	-0.000597184	-3.68307E-05
525	-0.89268213	1.384068995	-0.09519483	730	0.00386712	-0.0004198	-0.000025872
530	-0.809768076	1.45833629	-0.12206653	735	0.002685671	-0.000291736	-0.000017944
535	-0.689620194	1.498584815	-0.142514245	740	0.001853251	-0.000201467	-0.000012363
540	-0.53554014	1.50898709	-0.15698362	745	0.001278127	-0.000138559	-8.5748E-06
545	-0.34795458	1.49088951	-0.16578211	750	0.000891415	-9.65788E-05	-5.9876E-06
550	-0.12916182	1.44672063	-0.169577885	755	0.000630879	-6.82485E-05	-4.2505E-06
555	0.11928228	1.37991338	-0.169401065	760	0.000445708	-4.82894E-05	-2.9938E-06
560	0.39507816	1.2905718	-0.16574405	765	0.000314588	-3.45777E-05	-2.0511E-06
565	0.69274797	1.178470245	-0.15894077	770	0.000222854	-2.41447E-05	-1.4969E-06
570	1.0051998	1.04745006	-0.14953933	775	0.000158914	-1.77733E-05	-9.977E-07
575	1.32215514	0.90088377	-0.13791175	780	0.000113047	-1.25568E-05	-7.206E-07
580	1.63117509	0.744211405	-0.12469804				

Table 3: *Conversion Matrix from spectrum to RGB color vector*