GPU Rendering of the Thin Film on Paints with Full Spectrum

Roman Ďurikovič*
University of Saint Cyril and Metod, Trnava, Slovakia.

Ryou Kimura
Software Department, The University of Aizu, Japan

Abstract

Spectrum-based rendering uses spectral distributions instead of just three RGB colors for representation of light sources and surface properties in rendering equation. Since, spectrum has a value at every visible wavelength, the spectrum-based rendering gives much accurate color computation compared to RGB-based rendering and it give us opportunity to simulate wavelength dependent phenomena and effects caused by spectrum difference. We introduce a novel framework for spectrum-based rendering on GPU (Graphics Processing Unit) to compute local illumination. On this framework, the Phong reflectance model is implemented employing the spectral power distribution of light source and the spectral reflectance of surface simulating the color rendition of light sources and metamerism of surfaces. Multilayered thin film interference effects can also be handled within this framework with interactive speeds. Additionally, we propose the area light source defined by the spectral cube environment map and show the method for conversion of RGB environment map into a spectral one.

1 Introduction

Color-based and spectrum-based renderings are the frameworks to simulate the light transport within the scene given by rendering equation. Color-based rendering uses three colors red, green, and blue (RGB) to calculate the incoming radiance at camera point. Light sources and surface reflectance are represented by RGB instead of wavelength. It is the most popular framework for image synthesis.

The *RGB*-based rendering methods have several limitations, for example, the color rendition of light sources and metamerism of surfaces caused by the difference in spectrum can not be handled by *RGB*-based rendering.

*e-mail: roman.durikovic@fmph.uniba.sk. Also with Faculty of Mathematics, Physics and Informatics, Comenius University, Slovakia

Spectrum-based rendering uses spectral power distributions (SPDs) instead of RGB colors. This approach can support wavelength dependent effects by using intensities at each wavelength. The visible area of wavelength distributions of electromagnetic waves is from 380 nm to 780 nm. Sampling points' intervals from 1 nm to 5 nm will give the sufficient result. The strength of spectrum-based rendering is not only the simulation of wavelength dependent effects but also better color computation in the reflectance function. Spectrum-based rendering can also simulate wavelength dependent phenomena such as interference, dispersion, and diffraction. These effects require the calculation of amplitude and phase.

We introduce a novel framework for spectrum-based rendering on GPU (Graphics Processing Unit) to compute local illumination. Multilayered thin film interference effects, rendition of light sources and metamerism of surfaces can be handled within this framework with interactive speeds. We propose also the area light source defined by the spectral cube environment map and show the method for conversion of RGB environment map into a spectral one.

The framework is implemented on a common graphics card. The rendering is performed interactively by processing the spectral data on the fragment processor of GPU utilizing its parallelism.

1.1 Related Work

The spectrum-based rendering framework [6] consists of the part of reflectance function calculation and the color computation part. The reflectance function calculates the reflected wavelength distribution from the spectral power distribution (SPD) of light source. The color computation derives a color displayed on a monitor from a wavelength distribution incoming to the camera.

A light source has a SPD which is power distribution in wavelength of visible area. A SPD is measured by a spectrometer which measures irradiance or photon irradiance in units of W/m^2 and $photons/m^2$, respectively.

CIE standard illuminants (see Figure 1) are standard light sources. Illuminant named *A* has relative SPD of a tungsten filament whose temperature is approximately 2856 K. Illuminant *D*65 has relative SPD of average daylight with a color temperature of approximately 6500 K. Some types of light sources like fluorescent, mercury, and metal halide lamps have spikes in their SPDs.

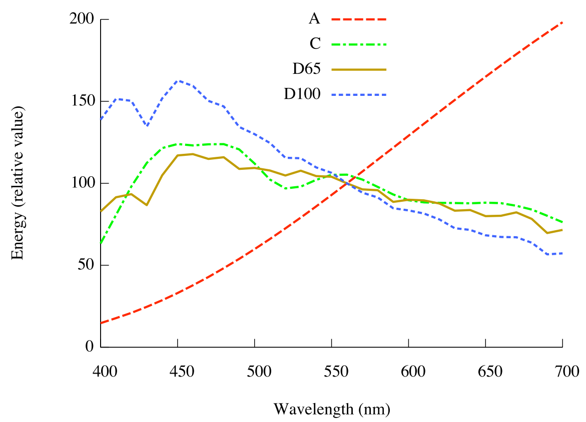

Figure 1. CIE standard illuminants.

The interference effect (or more precisely, amplitude splitting interference) is an optical effect observed in natural objects such as seashells and insects. Industrial products such as optical lens also demonstrate this effect. Light interference is caused by overlapping of light waves in thin films. When light enters a thin film, light waves are reflected on two sides of the film, if the light waves overlap, the energy of specific wavelengths are weakened. The implemented models simulating the interference effect caused by single or multiple thin film layers are introduced is [4]. The wavelength dependent effects depend on energy, amplitude, and phase at each wavelength [3].

2 Reflectance Function

Light reflection depends on surface properties such as refractive index, reflectance, absorbance, and transmittance. A spectral reflectance is ratio of energies of light reflection at each wavelength. Common spectral reflectances are collected in a Macbeth color chart.

2.1 Phong Reflectance Model

The Phong reflectance model [1] is a phenomenological reflectance model including diffuse and specular reflection. Though the Phong model is not physically-based, it can create BRDFs similar to plastic surfaces:

$$I_r(\lambda) = I_i(\lambda)\left[k_d R_d(\lambda)(\overrightarrow{\mathrm{N}}\cdot\overrightarrow{\mathrm{L}}) + k_s(\overrightarrow{\mathrm{R}}\cdot\overrightarrow{\mathrm{V}})^n\right], \quad (1)$$

where $I_r(\lambda)$ is the intensity of the reflected light, $I_i(\lambda)$ is the intensity of the incident light. Coefficients k_d and k_s control the ratio of the diffuse and specular reflection, respectively. $R_d(\lambda)$ is the spectral reflectance of the diffuse component of the surface, $\overrightarrow{\mathrm{N}}$ is unit surface normal vector. $\overrightarrow{\mathrm{L}}$, $\overrightarrow{\mathrm{R}}$, and $\overrightarrow{\mathrm{V}}$ are unit vectors in the direction of a light, the reflection of a light, and a camera, respectively. n controls the shininess of the surface.

Colored plastic is made of transparent plastic and color pigment. The specular and diffuse components of Phong model correspond to the reflection of the transparent plastic surface and color of the pigment, respectively. Therefore, $R_d(\lambda)$ is considered as the spectral reflectance of pigment in plastic (see Figure 9).

2.2 Interference Reflectance

Fresnel formulae give ratios of amplitudes of incoming light to reflected and transmitted light for parallel and perpendicular wave

$$\begin{cases} r_{\parallel} = \dfrac{n_t\cos\theta_i - n_i\cos\theta_t}{n_t\cos\theta_i + n_i\cos\theta_t} \\ r_{\perp} = \dfrac{n_i\cos\theta_i - n_t\cos\theta_t}{n_i\cos\theta_i + n_t\cos\theta_t} \\ t_{\parallel} = \dfrac{2n_i\cos\theta_i}{n_t\cos\theta_i + n_i\cos\theta_t} \\ t_{\perp} = \dfrac{2n_i\cos\theta_i}{n_i\cos\theta_i + n_t\cos\theta_t} \end{cases}, \quad (2)$$

where $r_{\parallel}$ and $r_{\perp}$ are ratios of amplitudes of reflected electric field vectors parallel and perpendicular to the plane of incidence, respectively. $t_{\parallel}$ and $t_{\perp}$ are ratios of amplitudes of transmitted electric field vectors parallel and perpendicular to the plane of incidence, respectively. Note that, the reflected and transmitted light is polarized in Eq. 2.

Because the interference effect is a wavelength dependent effect, light is described with electromagnetic wave with amplitude Φ_0:

$$\Phi(z,t) = \Phi_0 \exp i(kz - \omega t), \quad (3)$$

where z is the location, t is the time, k is the wave vector, and ω is the angular frequency.

If incoming light at the incident point is defined as $\Phi_i \exp(-i\omega t)$, reflected light will be $r_{0,1}\Phi_i \exp i(\delta - \omega t)$, were $r_{0,1}$ is the amplitude reflectance for the light incidence from medium 0 to medium 1, and δ is the phase difference along the reflected direction (see Figure 2).

The relation between reflected light $\Phi_r \exp i(\delta - \omega t)$ and incident light $\Phi_i \exp i(\delta - \omega t)$ is described by the following geometric series

$$\Phi_r \exp i(\delta - \omega t) = \{r_{0,1} + [(t_{0,1}t_{1,0})\, r_{1,2} \exp i\gamma] \\ \times \left[1 + (r_{1,2}r_{1,0}) \exp i\gamma + (r_{1,2}r_{1,0})^2 \exp i2\gamma + \cdots\right]\}$$

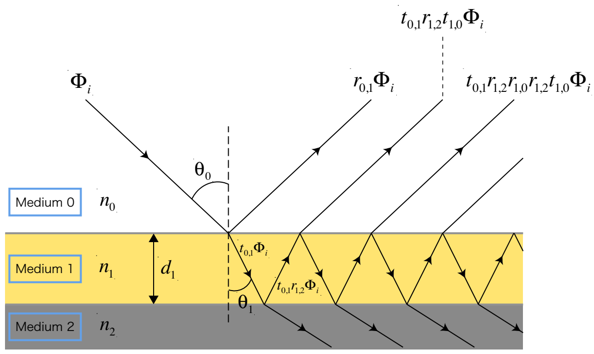

Figure 2. Multiple reflection on single layer surface.

$$\times \Phi_i \exp i\,(\delta - \omega t)\,, \tag{4}$$
$$\gamma = \frac{2\pi}{\lambda}\Delta, \tag{5}$$
$$\Delta = 2n_1 d_1 \cos\theta_1\,, \tag{6}$$

where $r_{i,j}$ and $t_{i,j}$ are the amplitude reflectance and transmittance for the light incidence from medium i to medium j from Eq. 2, γ is the phase difference, and Δ is the optical path difference. The geometric series in Eq. 4 can be simplified to

$$\Phi_r = \left\{ r_{0,1} + \frac{(t_{0,1}t_{1,0})\,r_{1,2}\exp i\gamma}{1-(r_{1,2}r_{1,0})\exp i\gamma} \right\} \Phi_i.$$

Taking into account Eq. 2 the entire ratio of the amplitude of reflected to incoming light is rewritten to

$$r_{0,2} = \frac{\Phi_r}{\Phi_i} = \frac{-r_{1,0} + r_{1,2}\exp i\gamma}{1-(r_{1,2}r_{1,0})\exp i\gamma}.$$

The energy reflectance of surface with a thin film layer is then given by

$$R_{0,2} = |r_{0,2}|^2\,. \tag{7}$$

Light transmitting a thin film is also influenced by the interference effect and thus energy transmittance $T_{0,2}$ derived from energy conservation law is :

$$T_{0,2} = 1 - R_{0,2}. \tag{8}$$

Finally, the light intensity exiting a surface is given by

$$I_r(\lambda) = I_i(\lambda)R_{0,2}(\lambda) + I_t(\lambda)T_{0,2}(\lambda), \tag{9}$$

where $I_i(\lambda)$ is the intensity of incident light, $I_t(\lambda)$ is the spectral reflecance of the substrat or the intensity of light from the backside of transparent surface.

3 Proposed Implementation on GPU

We implemented spectrum-based rendering on the fragment processor because of more flexibility for lighting and fast texture fetch. Temporary registers, constant registers, and texture fetch units on the fragment processor are used to simulate reflectance functions. Temporary registers are readable and writable while constant registers and texture units are read-only. The fragment processor reads a fragment from input registers and writes a computed fragment to output registers. Arithmetic instructions are executed with SIMD (Single Instruction Multiple Data). Modern fragment processors have the 128-bit 4-way SIMD architecture, so they can use 32-bit floating point. Our shader programs are written in Cg [5] and GeForce 6800 GT is a GPU used for our implementation.

3.1 Spectrum-based Rendering on the Fragment Processor

At the beginning of spectrum-based rendering, all resources must be set up. All data represented in spectral distribution, such as color matching functions, light sources, surface reflectance, are stored as 1D arrays in floating point textures with 32-bit per component. SPD data sampled at intervals greater that 5 nm may cause color shifts, so high precision data should be used for the accurate simulation of optical effects such as metamerism, and color rendition using spiky light sources. Visible wavelengths are from 380 nm to 719 nm interval, sampling it with 1 nm interval we get 340 spectral data points .

This large data can not be temporarily stored with a single pass operation and the fragment processor does not have the ability to write fragment data in memory with arbitrary addresses. Fortunately, random access memory reads are possible through texture fetching, but output addresses are fixed to specific pixels [2].

The local illumination model, proposed here consists of two parts, the computation of spectral reflectance function and the RGB color computation from SPD, refer to Figure 3.

3.1.1 Spectral Data Representation

In order to utilize the parallelism of the fragment processor we represent the spectral data by uniform discrete samples stored in an array.

The algorithm loops through all sampling points of spectrum stored in tristimulus (XYZ) values, evaluates the surface reflectance function and calculates the conversion integral from SPD to XYZ color representation (see Figure 4).

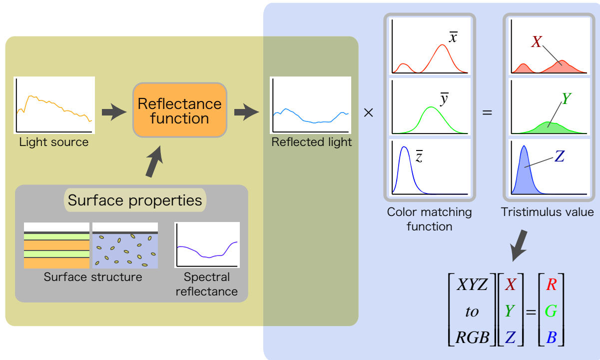

Figure 3. Proposed spectrum-based rendering algorithm on GPU.

On the fragment processor, 4 sampling points are loaded from textures, stored in temporary registers, and used as 4-way vectors in sequential order. By loading the quadruples of sampled data we assume that values at each wavelength are independent. This is true for many optical effects except for fluorescence and phosphorescence. Fluorescence is the phenomenon in which light energy at a wavelength is absorbed on a surface and then emission occurs at longer wavelengths therefore our algorithm can not simulate this effect.

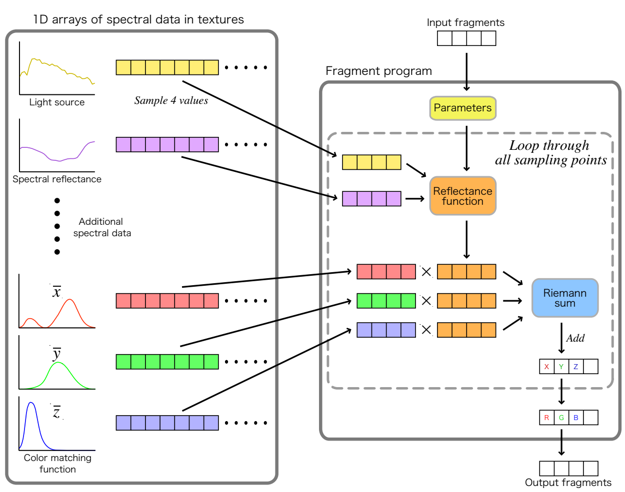

Figure 4. Spectrum-based rendering diagram on GPU.

3.1.2 Color Computation from SPD

After incident rays to the camera are calculated, the SPDs along those rays are transformed to RGB color space of a display. The transformation is based on the human visual system, consisting of two steps. First, the reflected light is multiplied by color matching functions, and integrated. Finally, tristimulus value XYZ is transformed to RGB color space.

CIE XYZ color matching functions are often used in industry instead of the CIE RGB model because CIE XYZ color matching functions, $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$, do not have negative ranges. Additionally, CIE XYZ color matching functions can be transformed to CIE RGB color matching functions by the linear transformation matrix. The $\bar{y}$ component of CIE XYZ is the photopic visibility function which determines the brightness of light. It has the maximal value at 555 nm, therefore greenish yellow is the brightest color per energy, (see Figure 5).

For given reflected SPD, $I(\lambda)$, the XYZ tristimulus values are derived by three integrals:

$$\begin{cases} X = k\int_{\lambda_{min}}^{\lambda_{max}} I(\lambda)\bar{x}(\lambda)d\lambda \\ Y = k\int_{\lambda_{min}}^{\lambda_{max}} I(\lambda)\bar{y}(\lambda)d\lambda \\ Z = k\int_{\lambda_{min}}^{\lambda_{max}} I(\lambda)\bar{z}(\lambda)d\lambda \end{cases} . \quad (10)$$

The scaling constant, k, is selected to set $Y = 100$ at the brightest point [1]:

$$k = \frac{100}{\int_{\lambda_{min}}^{\lambda_{max}} S(\lambda)\bar{y}(\lambda)d\lambda}, \quad (11)$$

where $S(\lambda)$ is the SPD of a light source.

Finally, XYZ values are transformed to RGB values. Color spaces for LCD and CRT displays consist of display-specific and standardized color spaces defined on a product-by-product basis. There are standardized color spaces like sRGB and Adobe RGB that are device-independent and

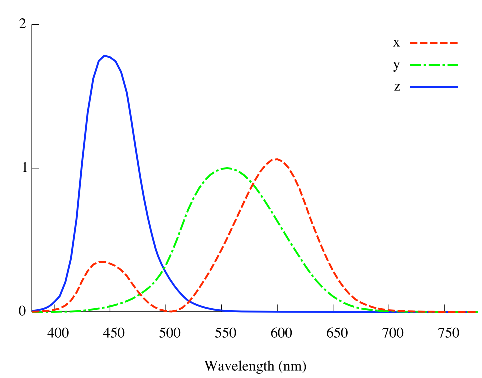

Figure 5. CIE XYZ color matching functions.

supported on general LCD and CRT displays. The following is the transformation matrix of XYZ to sRGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (12)$$

3.1.3 Integration Methods

Equations 10 and 11 include integration methods of spectral distributions for fast and accurate transformation of SPD to RGB color.

There are several methods that require small number of sampling points and also reduce computation costs while keeping accuracy. The Riemann integration will be used despite of large number of uniform sampling intervals required for accurate calculation. We can still obtain the real-time response for complex meshes.

4 Spectral Environment Illumination

Let us consider for a moment the illumination by the environment, in such case we will need to know the light SPD that is different for all incoming directions. Simple approach is to use and environment texture mapping. The environment map is a cube or spherical texture with real RGB values mapped on the cube or sphere around the rendered object. In order to have the spectral representation of environment we need to transform each RGB triplet from the environment map to a spectral distribution. Unfortunately, a single RGB triplet has infinite number of spectral distributions also known as *metamers*. Problem is to find a plausible metamer for each point in the environment map.

Our transformation method is simple and plausible for environment maps. A SPD for main light source is decided from the environment map as follows:

- If the environment is exterior, main light source will be the sun or standard light $D65$.
- For interior illumination, an artificial light such as incandescent or fluorescent lamps will be used.

Every pixel of the environment map occupied by the area of main illuminator has the SPD of know illuminator. For example the pixels occupied by the sun have the SPD of standard sun illuminator. For the remaining pixels the RGB values are converted to the SPD by the following equation

$$SPD(R,G,B) = (Rw_R MCC_R + Gw_G MCC_G + Bw_B MCC_B)\, SPD_{mls},$$

where R, G, B are the input color components, SPD_{mls} is the SPD of selected main light source, MCC_R, MCC_G, and MCC_B are the spectral distributions derived from the Macbeth color chart for the red, green and blue color, see Figure **??**. The weights w_R, w_G, w_B are determined by the following optimization problem:

For a given main light source with known spectra SPD_{mls} find the weights w_R, w_G, w_B minimizing the error $|RGB - RGB_{mls}|$, where RGB is spectral distribution $SPD(1,1,1)$ converted to the RGB color components, and RGB_{mls} is spectral distribution SPD_{mls} converted to the RGB color components.

5 Results

We demonstrate the implementation of proposed algorithms in our custom rendering system by using it to compute the real-time visualization of objects with different thin film thickness. Shown images are captured directly from computer screen.

Material structure in the first scene consists of chrome oxide layer with thickness of 270 nm and refractive index 2.7. The substrate layer is a black pigmented plastic with refractive index 1.45. Figure 6 demonstrates the ball illuminated by the SPD environment map.

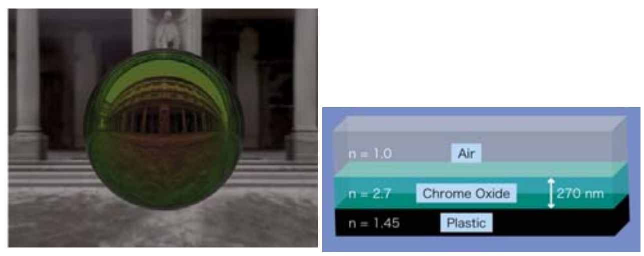

Figure 6. Chrome oxide thin film interference. Rendered ball (left) and material structure (right).

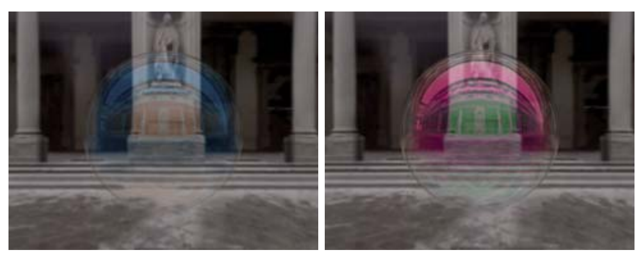

Figure 7. Titanium dioxide thin film interference. Rendered transparent ball with film thicknesses 150 **nm (left) and** 350 **nm (right).**

Figure 7 shows the transparent plastic surfaces with the thin film of Titanium Dioxide. The refractive indices of the thin film and substrate are 2.3 and 1.45, respectively. The film thicknesses of left and right images are 150 nm and 350 nm, respectively. Surfaces in Figure 8 have gradual changes

Figure 8. Chrome oxide thin film interference with increasing thickness from 100 **nm to** 400 **nm.**

of film thickness. The film thickness from left to right edge of cylinders changes from 100 nm to 400 nm, respectively. The refractive indices of the thin film and substrate are 2.7 and 1.45, respectively. Standard *D*65 light source is used in these simulations.

Simulation of color rendition using the orange illuminators *D*65 illuminator, incandescent lamp and low pressure sodium lamp demonstrate the effect when different materials look similar under similar illumination conditions, see Figure 9 bottom left.

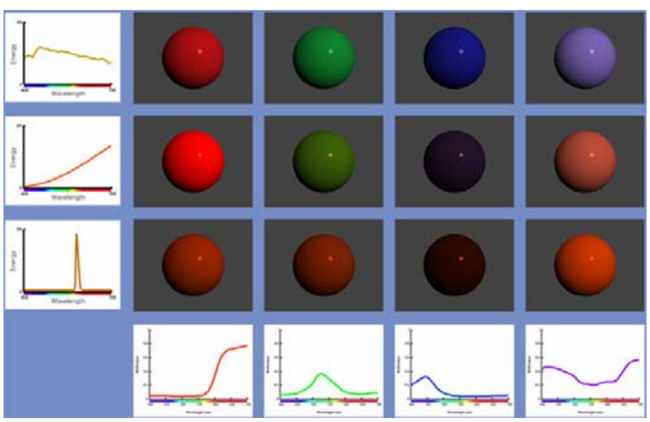

Figure 9. Simulation of color rendition using *D*65 **(top row), incandescent lamp (middle row), and low pressure sodium lamp (bottom row).**

6 Conclusion

In this paper we have presented a novel framework for high quality real-time spectral rendering on GPUs. By utilizing the parallelism of the fragment processor and effective management of sampled data we obtained the accurate simulation of light wave effects such as color rendition and interference while simultaneously obtaining a significant performance boost.

Realistic examples of transparent coatings used in paint industry were simulated at interactive speeds.

Acknowledgments

This research was sponsored by grants from the EU-FP6-MC–040681- APCOCOS and VEGA 1/3083/06.

References

[1] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics Principles and Practice*. Assison Wesley Publishing Company, Reading, Massachusetts, 2nd edition, 1996.

[2] M. J. Harris. Gpgpu: Beyond graphics. In *Eurographics Tutorial, (Eurographics Annual Conference 2004)*, http://developer.nvidia.com/object/gpgpu_beyond_graphics.html. Blackwell Publishing, 2004.

[3] E. Hecht. *Optics*. Addison Wesley Publishing Company, Reading, Massachusetts, 2nd edition, 1987.

[4] H. Hirayama, K. Kaneda, H. Yamashita, and Y. Monden. An accurate illumination model for objects coated with multilayer films. In *In Proceedings of EUROGRAPHICS 2000 Short Presentations*, pages 143–150. Eurographics, Blackwell Publishing, 2000.

[5] nVIDIA co. *Cg toolkit users manual a developers guide to programmable graphics release 1.2*. nVIDIA co., 2004.

[6] Y. Sun, F. D. Fracchia, M. S. Drew, and T. W. Calvert. A spectrally based framework for realistic image synthesis. *The Visual Computer*, 17(7):429–444, 2001.