Retro-rendering with Vector-Valued Light: Producing Local Illumination from the Transport Equation

David C. Banks and Kevin M. Beason

Florida State University, Tallahassee, USA

ABSTRACT

Many rendering algorithms can be understood as numerical solvers for the light-transport equation. Local illumination is probably the most widely implemented rendering algorithm: it is simple, fast, and encoded in 3D graphics hardware. It is not, however, derived as a solution to the light-transport equation.

We show that the light-transport equation can be re-interpreted to produce local illumination by using vector-valued light and matrix-valued reflectance. This result fills an important gap in the theory of rendering. Using this framework, local and global illumination result from merely changing the values of parameters in the governing equation, permitting the equation and its algorithmic implementation to remain fixed.

Keywords: local illumination, global illumination, fluorescence, spectral rendering, theory of rendering

1. INTRODUCTION

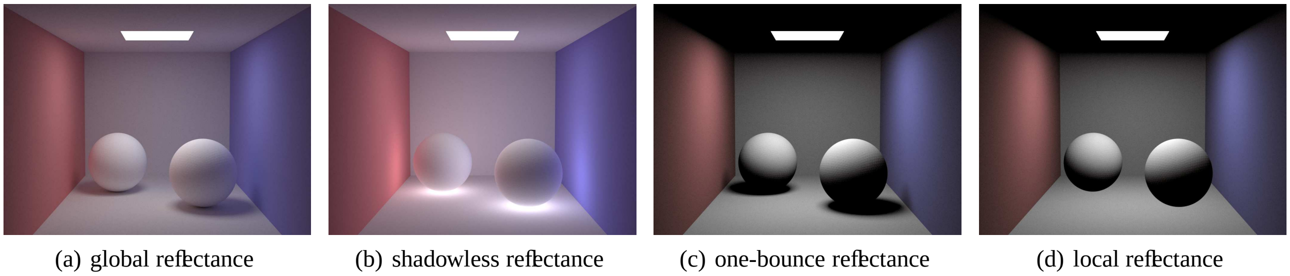

(a) global reflectance (b) shadowless reflectance (c) one-bounce reflectance (d) local reflectance

Figure 1. Figure 1: Globally illuminated scene exhibiting four different reflectance functions.

In the historical development of computer graphics, "local illumination" was the first and simplest illumination technique to produce three-dimensional shading in an image. The heart of the algorithm evaluates the exitant radiance $L_{\text{out}}(x, \omega_{\text{out}})$ in the direction ω_{out} from a point x on a surface according to the equation

$$L_{\text{out}}(x, \omega_{\text{out}}) = \sum_i f(x, \omega_i, \omega_{\text{out}})\, \hat{L}_{\text{in}}(x, \omega_i)\ \cos\theta \tag{1}$$

where the reflectance function f determines the appearance of the surface, the angle ω_i is measured from the point light at position p_i to the point x, and the angle θ is measured between the surface normal at x and the direction to p_i. The incident radiance in the direction ω_i from p_i to x is approximated by

$$\hat{L}(x, \omega_i) = \frac{I(p_i)}{r^2} \tag{2}$$

Further author information: (Send correspondence to D.C.B.)
D.C.B: E-mail: banks@cs.fsu.edu, Telephone: 1 850 644 5437
K.M.B.: E-mail: beason@cs.fsu.edu, Telephone: 1 850 644 5437

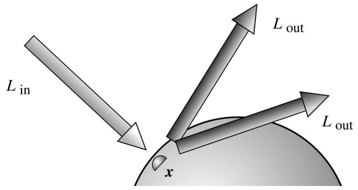


Figure 2. Incident radiance L_{in} strikes a point x on the surface and reflects with radiance $L_{\text{ out}}$ in directions ω_{out}

where $I(p_i)$ is the intensity at each point light p_i, and the distance $r = |p_i - x|$ is measured from the point p_i to x. Over the years, more sophisticated algorithms have been developed to produce increasingly realistic images (incorporating shadows and inter-reflections) by approximating the equation

$$L_{\text{out}}(x,\omega_{\text{out}}) = E(x,\omega_{\text{out}}) + \int_{\omega_{\text{in}}\in\mathbb{S}^2} f(x,\omega_{\text{in}},\omega_{\text{out}})\, L_{\text{in}}(x,\omega_{\text{in}}) \cos\theta \,\mathrm{d}\omega_{\text{in}} \tag{3}$$

for light transport,[1] where integration is over the sphere $\mathbb{S}^2$, L_{in} is the incident radiance, and the emitted radiance E is nonzero for luminaires. See Figure 2. In many domains of computational science, physical phenomena that obey governing equations are solved by numerical algorithms. The notion that rendering algorithms belong to the landscape of computational science has been articulated by several authors: the survey article by Christensen[2] describes how the task of rendering can be understood as the implementation of a numerical solver for Equation 3, either by gathering light into the camera (*e.g.* ray tracing, path tracing, and matrix-solver radiosity) or by shooting light from the luminaires (*e.g.* progressive radiosity and photon tracing). Kajiya's formulation of light transport[3] includes a "geometry term" $g(x_1,x_2)$ between points x_1 and x_2 that could be set to unity when x_2 lies on a luminaire, thus eliminating shadows. Can a term in equation 3 be likewise coerced into producing local illumination?

Although the two are rather similar, Equation 1 differs significantly from Equation 3 because the incident radiance L_{in} arriving at the point x generally differs from the approximation $\hat{L}_{\text{in}}$ due to scattering that may occur along the path P from p_i to x. Equation 1 is not intended to be a physically plausible description of light transport, but it leads to a fast implementation because integration along the path P can be neglected. Local illumination can therefore be viewed as a correct algorithm to evaluate the wrong equation, or as an incorrect algorithm to evaluate the right equation. But is it possible to interpret Equation 3 so that local illumination is actually the correct solution? If so, this would allow us to formulate a theory of rendering that incorporates the widely used (and widely taught) algorithm for local illumination as a correct numerical solver for a special case of the integral equation for light transport, as opposed to being a stand-alone algorithm that is merely suggestive of correct results. Under such a theory, the question "what lies between local illumination and global illumination?" would be meaningful because interpolating between parameters in an equation is well understood, whereas interpolating between algorithms is not.

2. VECTOR-VALUED LIGHT

The paper "Removing shadows from images"[4] shows how shadows in an image can be removed as a post-process; by contrast, we offer an interpretation of Equation 3 in which shadows are not even cast. We accomplish this via a bit of subterfuge inspired by locally-illuminated scenes rendered with OpenGL in which the point light sources are invisible to the camera and therefore do not appear in the image.

The problem with using a physically faithful renderer to generate a locally-illuminated shadowless image is that a physical object must be transparent in order to permit light to pass through it. Such a purely transparent surface is characterized by the reflectance function

$$f_{\text{transp}}(x,\omega_{\text{in}},\omega_{\text{out}}) = \frac{1}{\cos\theta}\ \delta(\omega_{\text{out}} - \omega_{\text{in}}) \tag{4}$$

(where δ is the Dirac delta distribution) which can be substituted into Equation 3 to produce the identity

$$L_{\text{out}}(x,\omega) = L_{\text{in}}(x,\omega) \tag{5}$$

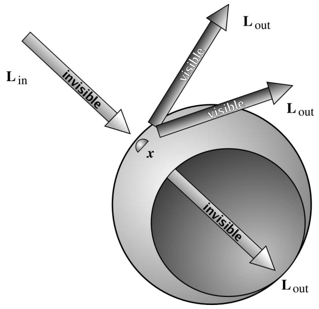

$\mathbf{L}_{\text{out}}$
visible
$\mathbf{L}_{\text{in}}$
invisible
visible
$\mathbf{L}_{\text{out}}$
x
invisible
$\mathbf{L}_{\text{out}}$

Figure 3. Incident invisible radiance $\mathbf{L}_{\text{in}}$ strikes a point x on a sphere (cut-away view). The surface transmits invisible light while reflecting visible light.

for transparency.[5] A perfectly transparent object casts no shadow, but it also reflects no light to the camera. So applying the reflectance f_{transp} does not quite produce local illumination, because it makes objects transparent both to the luminaires and to the camera. If, however, a transparent object could somehow generate "reflected" light while allowing the incident light to pass through transparently, then the object would be visible to the camera even while casting no shadow. In general, the "reflected" light obeys the same reflectance function (which we denote f_{opaque}) as the corresponding locally-illuminated surface; in particular, f_{diffuse} mimics an ordinary diffuse reflector by satisfying

$$f_{\text{diffuse}}(x, \omega_{\text{in}}, \omega_{\text{out}}) = \begin{cases} k_d & \text{if } \cos\theta > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where k_d is the diffuse coefficient. The drawback with combining f_{transp} with f_{opaque} is that the total amount of light increases with every scattering event, so the resulting image is unrealistically bright, especially where two surfaces interreflect.

Our solution is to treat light as a vector-valued quantity $\mathbf{L}$ having both an invisible and a visible component.

$$\mathbf{L} = \begin{bmatrix} L_{\text{invisible}} \\ L_{\text{visible}} \end{bmatrix} \tag{7}$$

Every pixel $\mathsf{p}[\mathsf{x},\mathsf{y}]$ in the image stores the vector-valued quantity $\mathbf{L}$ whose coordinates are the invisible and visible components of the light. The final displayed image shows only the second (visible) coordinate of the light at each pixel, namely the (inner) product

$$[0 \;\; 1] \; \mathbf{L} \tag{8}$$

of the vector $[0\ 1]$ with $\mathbf{L}$. Thus, light from the luminaires ultimately contributes to the light seen by the camera, but that contribution is made indirectly. The reflectance functions of the surfaces can be deliberately designed to produce local illumination while leaving the governing equation intact.

2.1. Local Illumination

We establish initial conditions for light transport by assigning vector-valued emittance

$$\mathbf{E} = \begin{bmatrix} E_{\text{invisible}} \\ E_{\text{visible}} \end{bmatrix} \tag{9}$$

to the luminaires. For OpenGL-style rendering where the light source is not visible in the image, we require

$$E_{\text{visible}} = 0 \tag{10}$$

so that the luminaire emits purely invisible light. To make the luminaire visible in the image, we simply require that

$$E_{\text{invisible}} = E_{\text{visible}} \tag{11}$$

so that the luminaire emits light into the scene and into the camera. The invisible light is transmitted perfectly through any object in its path (casting no shadow), but is reflected as visible light which is then seen by the camera. The visible light is absorbed by each surface in the scene. This behavior is summarized in the scattering diagrams below, with invisible light striking a point x and being transmitted and reflected, and with visible light being absorbed.

$$L_{\text{invisible}} \;\to x \begin{array}{l} \nearrow L_{\text{invisible}} \quad \text{(transmitted)} \\ \searrow L_{\text{visible}} \quad \text{(reflected)} \end{array} \tag{12}$$

$$L_{\text{visible}} \;\to x \qquad\qquad \text{(absorbed)} \tag{13}$$

The 2×2 matrix F_{local} given by

$$F_{\text{local}} = \begin{bmatrix} f_{\text{transp}} & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix} \tag{14}$$

organizes the transparent and reflective components describing the interaction between this invisible light and a surface so that the product

$$F_{\text{local}}\,\mathbf{L} = \begin{bmatrix} f_{\text{transp}}\,L_{\text{invisible}} \\ f_{\text{opaque}}\,L_{\text{invisible}} \end{bmatrix} \tag{15}$$

produces perfect transmission in the invisible component and opaque reflection in the visible component of the vector.

This matrix-valued interpretation of F_{local} may be inserted in the vector-valued equation for light transport

$$\mathbf{L}_{\text{out}} = \mathbf{E} + \int\limits_{\omega_{\text{in}}\in\mathbb{S}^2} F_{\text{local}}\,\mathbf{L}_{\text{in}}\,\cos\theta\,\mathrm{d}\omega_{\text{in}} \tag{16}$$

with emissive components $E = 0$ on surfaces of non-emissive objects and $E \neq 0$ on luminaires. The resulting image using F_{local} as the reflectance function is shown in Figure 1 (d). Note that the luminaire in this scene emits light that includes a nonzero visible component, otherwise it would not show up in the image.

Our proposed formulation of vector-valued light transport with matrix-valued reflectance has obvious similarity to fluorescent effects using spectral rendering[6789] in which a vector representation of light at different wavelengths (in particular, ultraviolet and visible) interacts with a surface via a reflectance matrix that converts incident light at one wavelength into reflected light re-radiated at another wavelength. In our model, the "wavelength" (or better, the pseudo-wavelength) λ is merely an abstract contrivance for interpreting the invisible light. The wavelength dependence can be stated explicitly using a reflectance matrix F_λ and spectral radiance $\mathbf{L}_\lambda$ as shown below

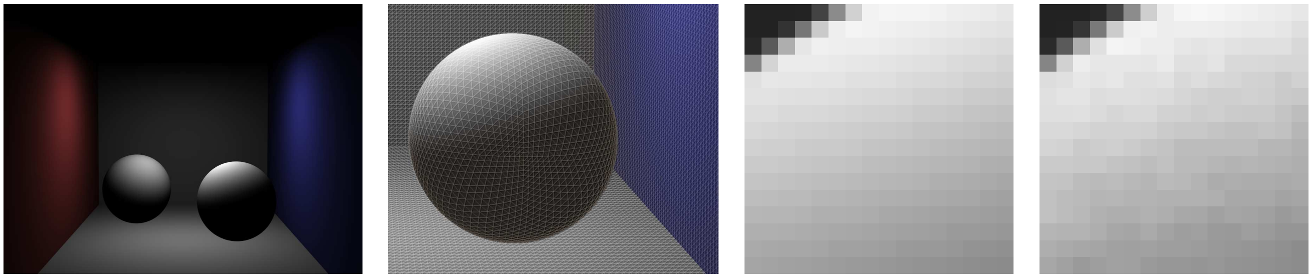
$$F_\lambda\,\mathbf{L}_\lambda = \left[\begin{array}{cc|cc|cc} f_{\text{transp}} & 0 & 0 & 0 & 0 & 0 \\ f_{\text{opaque}} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & f_{\text{transp}} & 0 & 0 & 0 \\ 0 & 0 & f_{\text{opaque}} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & f_{\text{transp}} & 0 \\ 0 & 0 & 0 & 0 & f_{\text{opaque}} & 0 \end{array}\right] \begin{bmatrix} \hat{L}_r \\ L_r \\ \hat{L}_g \\ L_g \\ \hat{L}_b \\ L_b \end{bmatrix} \tag{17}$$

where F_λ is a block-diagonal matrix whose 2×2 blocks are given by Equation 14 and $\hat{L}_\lambda$ denotes the invisible version of radiance L_λ having basis wavelength λ (whether red, green, or blue). The following sections present only the 2×2 blocks, as in Equation 14, from which the coresponding full-spectral reflectance matrix F_λ, as in Equation 17, can immediately be derived.

2.2. Global Illumination

To produce conventional global illumination for a scene lit by vector-valued light, the surfaces should reflect the "invisible" light in both the invisible and visible channels (in order to propagate into the scene while being visible to the camera), permit no forward scattering of light (otherwise objects become transparent), and absorb the visible light. The scattering diagram below illustrates this behavior.

$$L_{\text{invisible}} \;\to x \begin{array}{l} \nearrow L_{\text{invisible}} \quad \text{(reflected)} \\ \searrow L_{\text{visible}} \quad \text{(reflected)} \end{array} \tag{18}$$



(a) lighting with OpenGL (b) triangle mesh (detail) (c) OpenGL (detail) (d) path tracer (detail)

Figure 4. Comparison between local illumination (using OpenGL) and global illumination with the local diffuse reflectance function F_{local}.

The 2×2 matrix F_{global} given by

$$F_{\text{global}} = \begin{bmatrix} f_{\text{opaque}} & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix} \tag{19}$$

describes the interaction between this invisible light and a surface so that the product of F_{global} and the vector-valued light $\mathbf{L}$ properly reflects invisible light to the camera and into the scene. Figure 1 (a) shows the resulting globally illuminated image.

2.3. Hybrid Illumination

With a vector-valued interpretation of radiance $\mathbf{L}$ and a matrix-valued interpretation F of the reflectance function, hybrid illumination effects can be produced. For example, the multiple bounces of light in global illumination are free to proceed even when no shadows are cast; that is, shadows and inter-reflection may be de-coupled as shown in the scattering diagram below that models shadowless inter-reflecting surfaces.

$$L_{\text{invisible}} \rightarrow x \begin{array}{l} \nearrow L_{\text{invisible}} \quad \text{(reflected + transmitted)} \\ \searrow L_{\text{visible}} \quad \text{(reflected)} \end{array} \tag{20}$$

With this form of scattering, a surface casts no shadow because the invisible light is transmitted through it. This behavior is represented by the scattering function $F_{\text{shadowless}}$

$$F_{\text{shadowless}} = \begin{bmatrix} f_{\text{opaque}} + f_{\text{transp}} & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix} \tag{21}$$

which produces the shadowless image in Figure 1 (b).

Producing shadows without inter-reflection is accomplished by absorbing the invisible light while reflecting it into the visible channel; the visible light is absorbed by each surface in the scene and is merely available for the camera to collect. This behavior is illustrated in the scattering diagram below

$$L_{\text{invisible}} \rightarrow x \rightarrow L_{\text{visible}} \quad \text{(reflected)} \tag{22}$$

and is represented by the scattering function $F_{\text{oneBounce}}$

$$F_{\text{oneBounce}} = \begin{bmatrix} 0 & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix} \tag{23}$$

which produces the “one-bounce” image in Figure 1 (c).

3. COMPARISON TO OPENGL

We compared the result of global illumination using a diffuse F_{local} (rendered with a Monte Carlo path tracer for vector-valued light using Russian roulette for path termination) versus local illumination using pure diffuse reflectors (rendered with OpenGL as the reference algorithm) by creating a test scene containing two spheres lit from above. Precise head-to-head comparison is complicated by the fact that a point light source is almost never sampled by the path tracer, but is the only local source of light available within OpenGL. To roughly match the geometry of the luminaire in OpenGL and in the path tracer, we modeled the rectangular light source as a set of five point lights (placed in the center tile and the four corner tiles of a 3×3 checkerboard tiling of the luminaire) in the OpenGL scene while treating it as an area for the path tracer to sample.

OpenGL uses the Warn model for a point light having intensity I at position p and shining in direction $\mathbf{v}$. At position x,

$$I(x) = \frac{(\mathbf{u} \cdot \mathbf{v})^e}{a_0 + a_1 r + a_2 r^2}\, I(p) \tag{24}$$

where r is the distance from p to x, softening parameters a_0, a_1, and a_2 govern the quadratic intensity falloff, $\mathbf{u}$ is the unit vector from p to x, and e is an exponent controlling the angular distribution of the intensity. In order to match the emittance of the luminaires using OpenGL and using a path tracer, we set the parameters $a_0 = a_1 = 0$ (coefficient $a_2 = 0.0025$ serves as a scale parameter characterizing the spatial dimensions of the scene) and $e = 1$. The result is that OpenGL approximately matches the behavior of an area light source that emits with a cosine distribution.

Figure 4 shows a side-by-side comparison between a scene containing roughly $500,000$ triangles illuminated by OpenGL and by our path tracer using F_{local}. In the path-traced image the luminaire emits purely invisible light; it it not seen by the camera, but its reflection off of surfaces in the scene forms the image. We applied multi-pass rendering (generating 64 samples per pixel) to de-alias the OpenGL image. The path tracer generated $80,000$ samples per pixel. The difference in the images is largely due to the variance in the path tracer's evaluation of exitant radiance; in 4(d) a magnified view of the foreground sphere reveals this speckling. The per-pixel error e_{pixel} can be measured as $\sqrt{\frac{1}{3}\left((\Delta r)^2 + (\Delta g)^2 + (\Delta b)^2\right)}$ where Δr, for example, is the difference between the red components; thus $e_{\text{pixel}} = 1$ is the error between black $(0,0,0)$ and white $(1,1,1)$. The two images use the entire brightness range from black to white. No tone-mapping was applied. The mean per-pixel error between images 4(a) and 4(b) is 0.4%, or 1 part in 255. Our theoretical model introduces little if any bias compared to OpenGL: the difference between the average color of each image is less than 1 part in 255. The small error supports the claim that the vector-valued interpretation of light transport in Equation 16 is an effective model for local illumination because it accurately reproduces an image rendered by OpenGL.

4. CONCLUSION

During the history of computer graphics, local illumination developed as an ad-hoc technique that was later improved by deliberate derivation from the equation for light transport. Local illumination is still widely taught and widely used in graphics hardware; a theory of rendering should be able to explain it in terms of light transport. We have shown that a vector-valued interpretation of light transport supports such a theory of rendering. In particular, the matrix-valued reflectance function F can produce (1) local illumination, (2) global illumination, (3) shadows without inter-reflection, and (4) inter-reflection without shadows, all within a single algorithm. The corresponding values of F are summarized below.

$$\begin{aligned}
F_{\text{local}} &= \begin{bmatrix} f_{\text{transp}} & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix} \\
F_{\text{global}} &= \begin{bmatrix} f_{\text{opaque}} & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix} \\
F_{\text{shadowless}} &= \begin{bmatrix} f_{\text{opaque}} + f_{\text{transp}} & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix} \\
F_{\text{oneBounce}} &= \begin{bmatrix} 0 & 0 \\ f_{\text{opaque}} & 0 \end{bmatrix}
\end{aligned}$$

We implemented a path tracer (gathering light into the camera) that uses vector-valued light in order to demonstrate the effects that these reflectances produce. In particular, F_{local} yields an image that differs from an OpenGL rendering by only 1 part in 255 (1 bit per byte) on average, supporting the claim that light transport with F_{local} is an effective model for local illumination. We plan next to demonstrate vector-valued light transport within a light-shooting algorithm.

The authors gratefully acknowledge NSF grants #0083898 and #0430954. Thanks also to Andrew Glassner for helpful discussions about "magic light."

REFERENCES

1. D. S. Immel, M. F. Cohen, and D. P. Greenberg, "A radiosity method for non-diffuse environments," in *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 133–142, ACM Press, 1986.
2. P. H. Christensen, "Adjoints and importance in rendering: An overview," *IEEE Transactions on Visualization and Computer Graphics* **9**(3), pp. 329–340, 2003.
3. J. T. Kajiya, "The rendering equation," in *Computer Graphics (Proceedings of SIGGRAPH 86)*, **20**, pp. 143–150, Aug. 1986.
4. G. D. Finlayson, S. D. Hordley, and M. S. Drew, "Removing shadows from images," in *7th European Conference on Computer Vision (ECCV 2002) (Part IV)*, pp. 823–836, May 2002.
5. A. Glassner, *Principles of Digital Image Synthesis*, Morgan Kaufmann, 1995.
6. M. S. Peercy, "Linear color representations for full spectral rendering," in *Proceedings of SIGGRAPH 93*, *Computer Graphics Proceedings, Annual Conference Series*, pp. 191–198, Aug. 1993.
7. A. S. Glassner, "A model for fluorescence and phosphorescence," in *Fifth Eurographics Workshop on Rendering (June 1994)*, Stefan Haas *et al.*, ed., pp. 57–68, Springer-Verlag, 1994.
8. 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), pp. 429–444, 2001.
9. A. Wilkie, R. F. Tobler, and W. Purgathofer, "Combined rendering of polarization and fluorescence effects," in *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, pp. 197–204, June 2001.
10. H. W. Jensen, *Realistic Image Synthesis Using Photon Mapping*, A. K. Peters, 2001.