Adaptive spectral rendering with a perceptual control

Jean Claude Iehl and Bernard Péroche

Laboratoire d'Images de Synthèse de Saint Etienne
Ecole Nationale Supérieure des Mines de Saint Etienne
158, Cours Fauriel
F-42023 Saint Etienne cedex 2
jciehl@emse.fr peroche@emse.fr

Abstract. This paper proposes a spectral rendering method based on a ray tracing algorithm, whose purpose is a control of the perceptual error made. An adaptive representation of spectral data for light sources and materials is used, which leads, for each pixel, to the evaluation of an algebraic expression. The computations of complex spread for each visible wave, based on an extended photon map, are locally restricted by an adaptive evaluation of the expression. This method allows to simulate high quality physically-based pictures and, in particular, some specific phenomena like refracting objects, scattered caustics, ...
Key words: spectral rendering, ray tracing, adaptive representation, error control, caustics.

1 Introduction

The purpose of physically-based rendering algorithms is to generate realistic images, by simulating as efficiently as possible the spread and the behaviour of light interacting with materials in the scene.

One important element for this simulation is the full spectral character of light, which cannot be correctly taken into account by the usual representation of colors with three components. Several techniques have already been proposed for representing spectral data. Some of them are based on more or less sophisticated sampling methods ([Mey88], [DMCP94], [ZCB97]), and others use a projection on a set of basis functions ([RF91], [Pee93], [RP97]).

Another important point for the simulation is the independent propagation of each wave of the visible domain, especially in refracting objects. This aspect is often neglected because of computation times.

The aim of our work, which is an extension of [RP97], is to produce a rendering software taking into account the spectral representation of light sources and materials and the independent propagation of waves in the visible domain on one hand, able to achieve a control of the perceptual error made on the other hand.

The visible wavelength domain is split up into bundles of waves having similar behaviour and similar propagation. A progressive computation, based on an estimation of the color error made, allows to find a visually satisfactory solution, while executing much less computations than an exhaustive solution. We must remind that the idea of a progressive computation controlled by a perceptual error has been recently introduced in computer graphics, both in ray tracing ([BM95], [BM98]) and in radiosity ([GH97], [Mys98]).

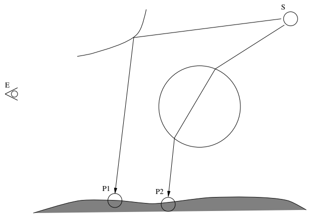

Fig. 1. Initial step for tracing LS^+D paths

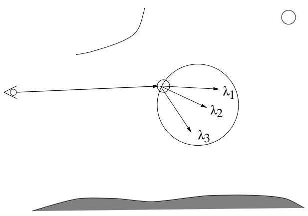

Fig. 2. Scattered gathering operator on an ES^+ path

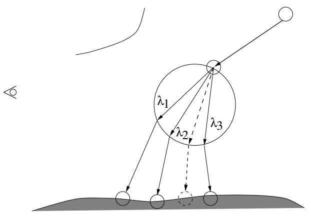

Fig. 3. Throwing again operator on a LS^+D path

We chose to work in a ray tracing environment [ZSP98] by using an adaptive representation of spectral data [RP97], a two pass method with an extended photon map [Jen96] and an estimation of the perceptual error which allows to control the evaluation of the expression of the luminous incident energy at each pixel. This expression will be called the *rendering expression* in the following of the paper.

The remainder of this paper is organized as follows. Section 2 presents the principle of our method. The adaptive representation is described in section 3, the evaluation of the rendering expression in section 4 and a spectral extension of the photon map in section 5. Some results are given in section 6 and a conclusion and some further developments are suggested in section 7. Details about the evaluation of operators of the rendering expression are given in an appendix.

2 Principle of the method

The computation of a picture is made in two passes. In the first one, the luminous energy emitted by light sources is followed according to specular LS^+D paths [Hec90]. In the second pass, the luminous incident energy at each pixel is gathered by explicitly building and then evaluating the expression which represents the interactions between light and materials with the help of operators.

- In the first pass, a photon map [Jen96] is built by following LS^+D paths coming from light sources (cf. figure 1). Each hit on a diffuse object produces the incident luminous flux according to the sampled path. During this pass, materials are supposed to have an "average" behaviour which is followed by all the waves, even if refractions appear on the sampled LS^+D path. The purpose of these average photons is to locate caustics in the scene.
- The second pass is a gathering stage from the eye. The rendering expression representing interactions between luminous energy and materials is built for each pixel. Two special cases may arise : a gathering path may encounter a refracting object (cf. figure 2) and/or an area of the scene with average photons (cf. figure 3). In each of these two cases, light paths must be sampled for each wavelength. But the waves that have to be sampled are not known at this stage of the computation. Decision will be taken at the time of the evaluation of the rendering expression. Special operators have just to be added to the expression, which will sample new refracted paths associated to the waves chosen at the time of the evaluation.

The evaluation at each pixel of the rendering expression is a progressive computation which iteratively subdivides the visible wavelength domain into intervals. The computation stops when the perceptual error associated with the current pixel falls under a given threshold. At each step of the computation, the interval producing the most important error is subdivided. This perceptual error is computed from operators building up the rendering expression, applied to adaptive representations of light sources and materials (cf. appendix).

This evaluation allows to concentrate the complex computation of tracing paths on the areas of the scene where transparent objects and caustics may be found and on the most important parts of the visible wavelength domain. The efficiency of the method arises from the reduction of the number of spectral computations and of rays cast, in relation to an exhaustive solution. A solution with an even sampling of the visible domain every $5nm$ would need to compute more than sixty monochromatic pictures. With our solution and on the examples described in section 6 with a natural light source, around 80% of the pixels are rendered with splitting up the visible wavelength domain into 4 or 6 intervals. Only pixels where refracting objects or caustics can be seen give rise to the sampling of new paths.

In fact, the evaluation of a rendering expression is a kind of rendering by visual importance ([DW95], [PP98]).

3 Adaptive representation of spectral data

The color of a point in the scene is the response of the visual system to the luminous energy coming from this point. This luminous energy results from interaction of light with the materials of the objects in the scene. Usually, these interactions are computed by evenly sampling the visible domain every $5nm$ for a good accuracy [Hun91]. Of course, this gives a very expensive rendering process. It's possible to reduce the number of samples by using a more sophisticated sampling scheme ([Mey88], [DMCP94], [ZCB97]), but this kind of solution does not fulfil our aim: with such a global treatment, the feasability of a local control of the error is lost. We have to choose a representation of spectral data which could adapt to the behaviour of materials and to the level of luminous energy, and thus which might allow to control the error and to guarantee the final result. For a maximal efficiency, it is preferable that computations are made directly on the representation itself and not on each sample, in order to make computations for a bundle of waves ([RP97], [Pee93]).

3.1 Representation of spectral data

We chose to represent spectral data by a hierarchical decomposition on a basis of scaling functions, which are usually associated to Haar wavelets: $f_i^j(x) = 1$ if $x \in \left[\frac{i}{2^j}, \frac{i+1}{2^j}\right]$ and 0 otherwise. The subdivision is dichotomous: the root of the decomposition is associated to the whole visible wavelength domain $[\lambda_{inf}, \lambda_{sup}]$. So each coefficient f_i^j of the decomposition fits an interval $[\lambda_a \ \lambda_b]$, where $\lambda_a = \lambda_{inf} + i\frac{\lambda_{sup}-\lambda_{inf}}{2^j}$ and $\lambda_b = \lambda_{inf} + (i+1)\frac{\lambda_{sup}-\lambda_{inf}}{2^j}$.

Representation on an interval. The coefficient associated to an interval represents an average value. As the purpose of the representation is to guarantee the color error associated to the luminous incident energy, we chose a value f_i^j which minimizes the

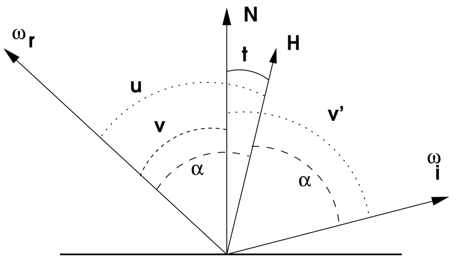

Fig. 4. Notations for BRDFs

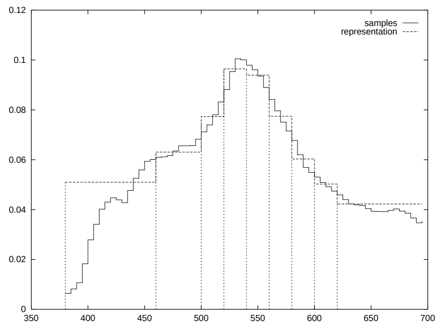

Fig. 5. 9 intervals for an error of 0.68 CIELAB

color error on interval $[\lambda_a \; \lambda_b]$:

$$f_i^j = \frac{\sum_{\bar{a}=\bar{x},\bar{y},\bar{z}} \int_{\lambda_a}^{\lambda_b} \bar{a}(\lambda)\, d\lambda \int_{\lambda_a}^{\lambda_b} f(\lambda)\bar{a}(\lambda)\, d\lambda}{\sum_{\bar{a}=\bar{x},\bar{y},\bar{z}} (\int_{\lambda_a}^{\lambda_b} \bar{a}(\lambda)\, d\lambda)^2} \quad (1)$$

where $\bar{x}(\lambda), \bar{y}(\lambda)$ and $\bar{z}(\lambda)$ are the color matching functions of the XYZ color space [WS82].

If we denote

$$\epsilon_{\bar{a}i}^j = \int_{\lambda_a}^{\lambda_b} (f(\lambda) - f_i^j)\bar{a}(\lambda)\, d\lambda \text{ , with } \bar{a} = \bar{x}, \bar{y}, \bar{z} \quad (2)$$

the color error associated to interval $[\lambda_a \; \lambda_b]$ is $\varepsilon_i^j = \sqrt{(\epsilon_{\bar{x}i}^j)^2 + (\epsilon_{\bar{y}i}^j)^2 + (\epsilon_{\bar{z}i}^j)^2}$.

In conclusion, an interval is represented by its average value f_i^j and three associated errors $\epsilon_{\bar{x}i}^j, \epsilon_{\bar{y}i}^j, \epsilon_{\bar{z}i}^j$.

Representation of materials: a spectral BRDF. The local behaviour of a material is described, for a given wavelength, by a function called BRDF. To cover up the lack of measures of spectral BRDFs, it is usual to suppose that such a function is separable into a geometric term (analytic or measured) and a spectral one.

Analytic models from C. Schlick [Sch94] and G. Ward [War92] introduce this separation. We used the model defined in [Sch94]: $f_{r\lambda}(t, u, v, v') = S_\lambda(u)D(t, v, v')$, where $D(t, v, v')$ is the geometric term (see figure 4 for notations) and $S_\lambda(u) = C_\lambda + (1 - C_\lambda)(1 - u)^5$ is the spectral term (C_λ defines the spectral behaviour of the material at normal incidence).

Opaque materials are represented by their reflectance. Transparent materials are defined through an index of refraction and an absorption coefficient. Reflectances and indices of refraction are available for a great number of materials (glasses for example).

NOTE: On figure 5, we may see the cutting up of a spectral data obtained with our adaptive evaluation.

4 Rendering expression

Reflected radiance $L_\lambda(P, \vec{\omega_r})$ at point P in direction $\vec{\omega_r}$ is the result of the propagation of light and its interaction with encountered materials. This radiance may be decomposed into several components, where each component may be computed by using a specific method. Following [ZSP98], we may write :

$$\begin{aligned} L_\lambda(P, \vec{\omega_r}) &= L_{e\lambda}(P, \vec{\omega_r}) + L_{spec\lambda}(P, \vec{\omega_r}) + L_{dir\lambda}(P, \vec{\omega_r}) \\ &+ L_{caust\lambda}(P, \vec{\omega_r}) + L_{ind\lambda}(P, \vec{\omega_r}) \end{aligned}$$

where

- $L_{e\lambda}(P, \vec{\omega_r})$ is the self-emitted radiance at point P along direction $\vec{\omega_r}$;
- $L_{spec\lambda}(P, \vec{\omega_r})$ is the reflected radiance along direction $\vec{\omega_r}$ according to specular reflection and refraction;
- $L_{dir\lambda}(P, \vec{\omega_r})$ is the reflected radiance coming directly from light sources without having been reflected or refracted;
- $L_{caust\lambda}(P, \vec{\omega_r})$ is the reflected radiance coming from light sources according to LS^+D paths;
- $L_{ind\lambda}(P, \vec{\omega_r})$ is the reflected radiance coming from light sources after having been speculary reflected.

Each component corresponds to a type of light path, which allows not to count twice some type of luminous energy. $L_{caust\lambda}(P, \vec{\omega_r})$ is computed by density estimation (cf. section 5.2). $L_{ind\lambda}(P, \vec{\omega_r})$ is either a constant ambient term or a term computed with a vector model [ZSP98] called the indirect light vector and $L_{dir\lambda}(P, \vec{\omega_r})$ is computed with the help of the direct light vector [ZBP99].

The rendering process builds up a rendering expression for the luminous energy gathered at each pixel. For each interaction of a ray with a material, several operators are added to the expression. They fit with the different terms of the rendering equation: direct, reflected or refracted energy at the given point.

When a BRDF is involved in the computation of a luminous energy, the geometric term must be evaluated in order to build up its spectral representation. We might have computed the representations of all the intervals of the decomposition, but it's more efficient to create an unary operator which will compute the representations of the needed intervals.

Operators play a very important role. The simulation of the independent spread of waves of the visible wavelength domain is based on an operator, so as the estimation of $L_{caust\lambda}$. Each such operator must return a color error taking into account not only the cutting up of the visible wavelength domain, but also the error made when sampling the light paths.

5 The scattered photon map

In several cases, it is necessary to find the exchange of luminous energy between two points, that is to say to find all light paths linking the two given points. There is no practical solution to this problem, especially when all the waves of the visible domain are taken into account and when refracting objects can be found in the scene. However, it is possible to estimate this luminous energy. For example, Jensen ([Jen96], [Jen97]), in a first pass, emits particles of energy from the light sources and keeps track of the hits on the objects in the scene. A second pass uses the density of the hits around a given point to estimate the reflected radiance at this point. The efficiency of the method is based on a data structure allowing a quick retrieval of hits, the k-d tree [Ben90] and on density estimation techniques [Sil86].

Our solution is inspired by Jensen's work ([Jen96], [Jen97]), but with some modifications to take into account the spectral aspect of our work.

Photons are emitted by light sources in the scene. They are traced according to the light path associated to their carrier wave. But, unlike [Jen97], when a photon hits an object, reflected and eventually refracted paths are systematically traced. Only LS^+D paths are sampled. Hits are thus stored only on non specular objects. Density estimation is only used to build up the incident radiance according to these paths; the other paths are sampled by a usual ray tracing (cf. section 4).

The energy associated to each photon is the incident luminous flux attenuated by each material encoutered.

5.1 Construction of the photon maps

This construction is made in two steps. The first one is a pre-computation, and the second one is an adaptive computation.

Pre-computation. This step is made before the beginning of the computation of the picture. Its purpose is to locate areas where caustics will appear. Average photons are traced according to the directions of specular or refracting objects visible from the light source and to the average behaviour of materials (cf section 2).

Progressive construction. This step is made during the computation of the picture. Let P be a point in the scene. We want to compute the reflected radiance at point P for a bundle of waves through density estimation. The average photons with the shortest distance to P show what are the LS^+D paths to be sampled again. For this new propagation, the behaviour associated to the materials is settled for the bundle of waves to be evaluated. The new paths are traced from the first refraction point. New photons are laid down in the photon map associated to the bundle of waves. In order to evaluate (by density estimation) the reflected radiance for this bundle of waves, new photons with the shortest distance to P are looked up.

Average photons gathered and not affected by refraction are thus propagated only once, during the pre-computation step. As the geometric part of reflection is constant on the visible wavelength domain, it's sufficient to revalue the rendering expression associated to the incident luminous flux for estimating the reflected radiance in the given interval.

The concentration of the photons laid down during the computation increases in the most important areas of the scene. Only photon maps connected to evaluated bundles of waves are built up.

5.2 Estimation of reflected radiance

The hits stored in the photon maps allow to find the radiance incident at a point P according to LS^+D paths. The technique used is similar to the one of [Col97] and [Jen97]. The n photons with the shortest distance to P are found in the photon map

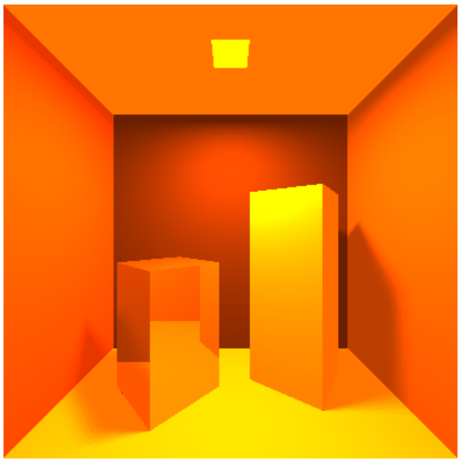

Fig. 6. Cornell box with a sodium light source

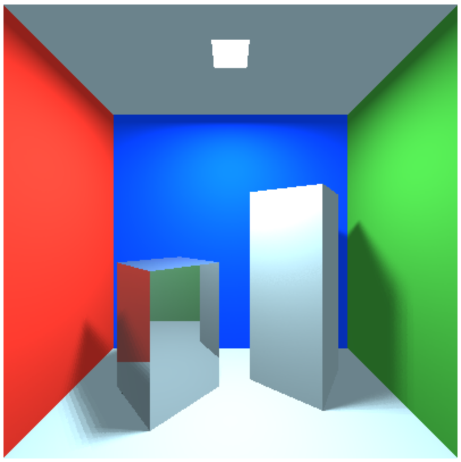

Fig. 7. Cornell box with a D6500 light source

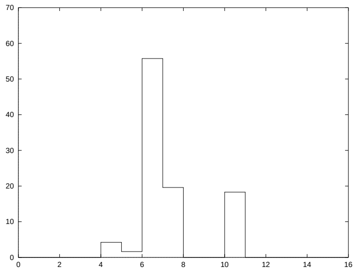

Fig. 8. Statistics for intervals

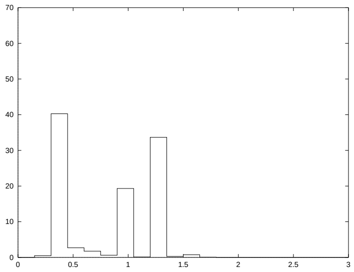

Fig. 9. Statistics for CIELAB errors

connected to the studied interval. As photons are not laid down everywhere in the scene, the search area is bounded. An estimation kernel K allows to ensure the continuity of the rebuilt function [Sil86]. Radiance reflected at point P along direction $\vec{\omega_r}$ is given by:

$$L_{caust\lambda}(P, \vec{\omega_r}) = \sum_{i=1}^{n} K\left(\frac{||\overrightarrow{Px_i}||}{r}\right) f_r\,(x_i, \vec{\omega_i} \to \vec{\omega_r}) \frac{\Delta\Phi_i(x_i, \vec{\omega_i})}{\pi r^2}$$

where x_i is photon i, $\vec{\omega_i}$ its incidence direction, $\Delta\Phi_i(x_i, \vec{\omega_i})$ its associated flux and r is the distance between P and the most distant photon in the search area.

6 Results

Figure 7 shows a classical Cornell box with a D6500 light source corresponding to a natural lighting. Materials are built with some reflectances extracted from the ColorChecker [MMD76] and a simple isotropic BRDF [Sch94]. Figure 6 shows the same scene lit with a sodium light source which emits some energy only in a very narrow band of the visible wavelength domain. The same scene, lit with a sodium light source and

Table 1. distribution of photons (interval (j,i) fits scaling function f_i^j)

# of photons	interval	# of photons	interval
37,004	(0,0)		
19,887	(1,0)	19,834	(2,3)
19,896	(1,1)	16,802	(3,2)
16,798	(2,0)	16,810	(3,3)
16,804	(2,1)	19,757	(3,4)
19,831	(2,2)	19,759	(3,5)

computed with only three RGB samples, shows many aberrations like green highlights on the blue wall lit by the orange light source.

Arrangement of the number of intervals (resp. of errors in the CIELAB color space) for figure 7 are shown on figure 8 (resp. 9). The rendering process splits up about 80% of the pixels into 6 or 7 intervals, for a CIELAB threshold equal to 3. As can be seen on figure 9, all the errors are very lower than the given threshold.

Figure 10 shows a refracted caustic computed with 37,004 average photons. The ring is glass made. The glass is defined by its Abbe number 437907 (the index of refraction is 1.437 at $587nm$) and its absorption coefficient is constant. The rendering process generates 186,178 additionnal photons, the distribution per interval of which is given in table 1. Rendering time is 1h39' and 65Mb of memory are filled up with 223,182 photons. Density estimation uses the 50 closest photons.

Figure 11 shows the same scene but this time the ring is built with a specular BRDF. Rendering time is 3'26 with the same parameters than for figure 10 and only 8,413 average photons are used. The light source is sampled exactly as above. Because of the systematic tracing of refacted rays, constructing the photon map for figure 10 laid down more than 37,000 average photons. The huge difference of rendering time is explained by the need of tracing new paths through the glass. The worst case is the one of refracted photons seen through the glass. Density estimation uses the 50 closest photons, that is to say 50 paths must be sampled for rendering each bundle of waves during a gathering step.

Figure 12 shows the decomposition of light by a prism. Rendering time is only 6'50. There are 40,558 average photons. 251,451 photons are used for a memory load of 75Mb. The photons are only laid down on the caustic on the wall and on the caustic between the prism and the wall. There is an hidden caustic on the rearside of the rightest green blocker. The rendering process spent most of its time to compute the small caustic on the wall.

7 Conclusions and further developments

In this paper, we have developed an adaptive spectral rendering method which allows to take into account the spectral character of light sources, materials and waves. The main advantage of the method is its ability to achieve a perceptual control of the color error made. Thus the computing effort to produce a picture can be focused on perceptually relevant features. The accuracy can still be chosen by the user. Furthermore, this method allows to compute high quality physically-based images with, for example, refracting objects and scattered caustics in the scene.

But there is still some work to do to improve our method. The perceptual error is currently only a rough estimate of the CIELAB error. We plan to integrate a vision model (cf. for example [TH94], [Wat98], [TAP98]) in order to improve the control on

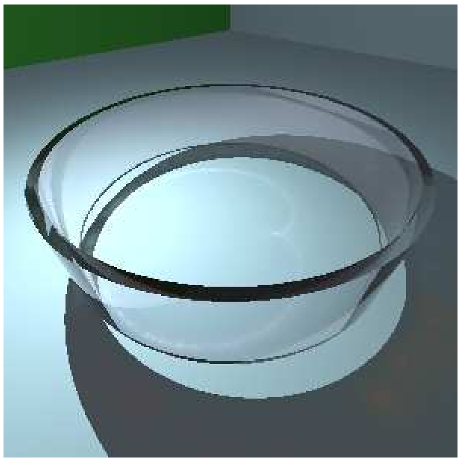

Fig. 10. A refracted caustic

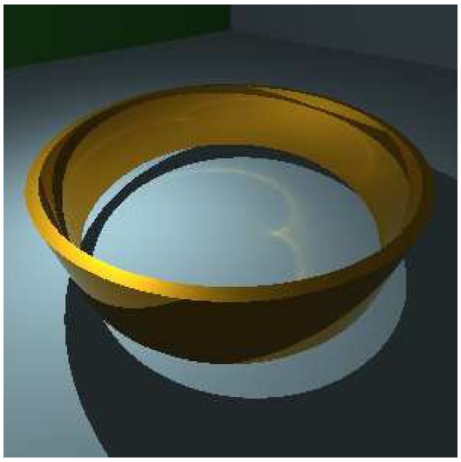

Fig. 11. A reflected cautic

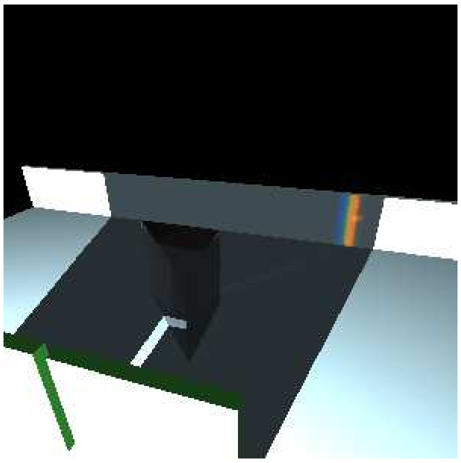

Fig. 12. A prism

the rendering process. The use of such a model would allow to go on from a control located on a pixel to a control extended over an area. We could also extend our method to colored glasses and thin layers, and use paraxial optic for throwing again photons in not too scattered materials. Finally, we could use subdivision caches for the structure of rendering expressions, in order to reduce the number of initial subdivisions and thus the number of stored photons.

References

Alb98. Stéphane Albin. Distance entre images. *Rapport de DEA, Ecole des Mines de Saint-Etienne*, 1998.

Ben90. Jon Louis Bentley. K-d trees for semidynamic point sets. In *Proceedings of the 6th Annual Symposium on Computationnal Geometry*, pages 187–197, June 1990.

BM95. Mark R. Bolin and Gary W. Meyer. A frequency based ray tracer. *Computer Graphics*, 29(3):409–418, August 1995.

BM98. Mark R. Bolin and Gary W. Meyer. A perceptually based adaptative sampling algorithm. *Computer Graphics*, 33(3):299–309, July 1998.

Col97. Steven Collins. *Wavefront tracking for global illumination solutions*. PhD thesis, Trinity College, Dublin, January 1997.

DMCP94. P. M. Deville, S. Merzouk, D. Cazier, and J. C. Paul. Spectral data modeling for a lighting application. *Computer Graphics Forum*, 13(3):97–106, 1994. Eurographics '94.

DW95. Philip Dutre and Yves D. Willems. Potential-driven monte carlo particle tracing for diffuse environments with adaptive probability density functions. In *Rendering Techniques '95 (Proceedings of the 6th Eurographics Workshop on Rendering)*, pages 306–315. Springer-Verlag, 1995.

GH97. S. Gibson and R. J. Hubbold. Perceptually-driven radiosity. *Computer Graphics Forum*, 16(2):129–141, 1997.

Hec90. Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics*, 24(4):145–154, August 1990.

Hun91. R. W. G. Hunt. *Measuring Colour, 2nd Ed.* Ellis Horwood, 1991.

Jen96. Henrik Wann Jensen. Global illumination using photon maps. In *Rendering Techniques '96 (Proceedings of the 7th Eurographics Workshop on Rendering)*, pages 22–31, 1996.

Jen97. Henrik Wann Jensen. Rendering caustics on non-lambertian surfaces. *Computer Graphics Forum*, 16(1):57–64, 1997.

Mey88. Gary W. Meyer. Wavelength selection for synthetic image generation. *Computer Vision, Graphics, and Image Processing*, 41(1):57–79, January 1988.

MMD76. C.S. McCamy, H. Marcus, and J. G. Davidson. A color-rendition chart. *Journal of Applied Photographic Engineering*, 2(3):95–99, July 1976.

Mys98. Karol Myszkowski. The visible differences predictor: applications to global illumination problems. In *Rendering Techniques '98*, pages 223–236, June 1998.

Pee93. Mark S. Peercy. Linear color representations for full spectral rendering. In *Computer Graphics*, volume 27, pages 191–198, August 1993.

PP98. Ingmar Peter and Georg Pietrek. Importance driven construction of photon maps. In *Rendering Techniques '98 (Proceedings of the 9th Eurographics Workshop on Rendering)*, pages 269–280. Springer-Verlag, 1998.

RF91. Maria Raso and Alain Fournier. A piecewise polynomial approach to shading using spectral distributions. In *Proceedings of Graphics Interface '91*, pages 40–46, June 1991.

RP97. Gilles Rougeron and Bernard Péroche. An adaptive representation of spectral data for reflectance computations. In *Rendering Techniques '97*, pages 127–138. Eurographics, June 1997.

Sch94. C. Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum*, 13(3):C/233–C/246, 1994.

Sil86. B. W. Silverman. *Density Estimation for statistics and data analysis*. Chapman et Hall, 1986.

TAP98. Christopher C. Taylor, Jan P. Allebach, and Zigmunt Pizlo. The image fidelity assessor. In *Procedings of the 1998 IS&T Image Processing, Image Quality, and Image Capture Systems Conference*, pages 223–236, May 1998.

TH94. Patrick Teo and David Heeger. Perceptual image distortion. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume 2, pages 982–986, November 1994.

War92. Gregory J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics*, 26(2):265–272, July 1992.

Wat98. A. B. Watson. Toward a perceptual video quality metric. *Human Vision and Electronic Imaging III*, 3299:139–147, May 1998.

WS82. G. Wysecki and W. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley and Sons, New York, 1982.

ZBP99. J. Zaninetti, P. Boy, and B. Péroche. An adaptive method for area light sources and daylight in ray tracing. In *the Proceedings of Eurographics '99*, 1999. to appear.

ZCB97. E. Zeghers, S. Carré, and K. Bouatouch. Error-bound wavelength selection for spectral rendering. *The Visual Computer*, 13(9+10):424–434, 1997.

ZSP98. J. Zaninetti, X. Serpaggi, and B. Péroche. A vector approach for global illumination in ray tracing. *Computer Graphics Forum (Proceedings of Eurographics'98)*, 17(3):149–158, September 1998.

A Appendix: Operators for the rendering expression

An operator is represented by a tree whose root specifies the operator and the sub-trees the operands. The purpose of this representation is to find the result of the operator for a given interval, without doing the computation for all the samples. We must find the representation of the resulting interval (i.e. its average value and the associated errors) from the representation of the intervals of the operands. This representation will be found during the adaptive evaluation.

A.1 Adaptive evaluation of an operator

The evaluation of an operator must determine one representation setting a color error lower than a given threshold. In order to limit the number of intervals, we chose to refine the cutting up by subdividing the interval for which the error is the most important. To obtain the result of a given operator, we first take into account the whole visible wavelength domain. The evaluation is then a progressive computation where an interval is subdivided while the error associated to the representation is greater than a given threshold.

We will now give some pieces of information about the evaluation of the various operators needed to perform a rendering computation.

Evaluation of a sum and of a product by a coefficient. Let $S1(\lambda)$ and $S2(\lambda)$ be two spectral power distributions, $S1_i^j$ and $S2_i^j$ their average values, $\epsilon_{S1\bar{a}i}^j$ and $\epsilon_{S2\bar{a}i}^j$ their color errors on interval $[\lambda_a,\ \lambda_b]$. As equations (1) and (2) are linear, $S1(\lambda) + S2(\lambda)$ is represented by $S1_i^j + S2_i^j$ and $\epsilon_{S\bar{a}i}^j = \epsilon_{S1\bar{a}i}^j + \epsilon_{S2\bar{a}i}^j$ and $c \cdot S1(\lambda)$ by $c \cdot S1_i^j$ and $c \cdot \epsilon_{S1\bar{a}i}^j$.

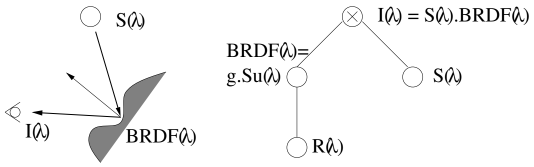

Fig. 13. a rendering expression

Evaluation of a product. Let $S(\lambda) = S1(\lambda)S2(\lambda)$. Thanks to equation

$$\begin{aligned} X &= \int_{\lambda_a}^{\lambda_b} (S1(\lambda) - S1_i^j)(S2(\lambda) - S2_i^j)\bar{x}(\lambda)\, d\lambda \\ &+ S1_i^j \epsilon_{S2\bar{x}i}^j + S2_i^j \epsilon_{S1\bar{x}i}^j + S1_i^j S2_i^j \int_{\lambda_a}^{\lambda_b} \bar{x}(\lambda)\, d\lambda \end{aligned}$$

and by neglecting the first term, we represent $S(\lambda)$ by $S_i^j = S1_i^j S2_i^j$ and $\epsilon_{S\bar{a}i}^j = S1_i^j \epsilon_{S2\bar{a}i}^j + S2_i^j \epsilon_{S1\bar{a}i}^j$

Evaluation of a BRDF. For a given geometry, the spectral term of a BRDF is given by: $S_u(\lambda) = C(\lambda) + (1 - C(\lambda))f_u$. Let C_i^j be the average value and $\epsilon_{C\bar{a}i}^j$ be the color errors associated to $C(\lambda)$. A representation of $S_u(\lambda)$ is given by $S_{ui}^j = C_i^j(1 - f_u) + f_u$ and $\epsilon_{S_u i}^j = (1 - f_u)\epsilon_{C\bar{a}i}^j$.

A.2 Evaluation of an expression

The strategy used is similar to the one used for an operator. We just give an example of such an evaluation. The expression given figure 13 corresponds to the picture produced by the reflection on a wall (material obtained from reflectance green-14 of the ColorChecker [MMD76] and from a simple isotropic BRDF [Sch94]) lit by a D6500 area light source. The representation given in figure 5 for a CIELAB threshold equal to 3 requires 9 intervals and produces an error equal to 0.68 in the CIELAB color space.

A.3 Choice of a threshold for the color error

The color errors of representations are given in the XYZ color space. To impose that the color computed with our method is perceptually the same than the one computed by an even sampling, a perceptual color space like CIELAB should be used. We subdivide the XYZ unit cube in regular cells and estimate, for each cell, the XYZ error corresponding to a given CIELAB threshold.

Tests made in our laboratory [Alb98] showed that a CIELAB error lower than 5 is non perceptible in the usual conditions of use of a workstation.