An adaptive representation of spectral data for reflectance computations

Gilles Rougeron and Bernard Péroche

Ecole des Mines de Saint Etienne
Centre Simade, LISSE
158, Cours Fauriel
42023 Saint Etienne Cédex 2, France
email : rougeron@emse.fr peroche@emse.fr

Abstract

This paper deals with the representation of spectral data so as to control the colorimetric error committed during rendering computations. These data are projected on a set of hierarchical basis functions called scaling functions leading to a representation by means of binary trees. An adaptive algorithm is proposed in which refinement and merge steps managed by an estimation of the error made in the XYZ color space allows to control the representation of spectra

Keywords: Spectral data, color representation, tristimulus values.

1 Introduction

The main aim of realistic image synthesis is to capture the visual appearance of modeled scenes. To reach such a goal, physically-based rendering methods are used which make possible to simulate the propagation of light energy in scenes. These methods must account for the full spectral character of the lights and surfaces within a scene. But in computer graphics, image generation uses the RGB colorimetric system for output for some display device such as a CRT monitor.

Thus the problem we are faced with is the following: At the time of a spectral rendering computation, how do we obtain RGB values at a lower cost, in such a way that psycho-visual errors committed during calculations are controlled? The purpose of our algorithm is to keep the error under some threshold, and to avoid as far as possible unnecessary computations. In fact, we shall solve this problem for XYZ tristimulus values because the XYZ color space is standardized and independent of any display device, and because XYZ tristimulus values can easily be related to RGB values.

Our research is made in the framework of a ray tracing environment [Roe93]. A spectral ray tracer was developed in our laboratory [Mai96], the visible wavelength domain being evenly sampled every 10 nm. This gives a very expensive rendering process and one of our goals is to achieve an adaptive sampling allowing to deal with less samples than in the previous work.

Our work will be limited to reflections, but transmissions could be handled in a similar way. Formally, if $S(\lambda)$ is the spectral power distribution of a light source, the spectrum reflected from a material with reflectance $R(\lambda)$ under normal incidence is $S_r(\lambda) = S(\lambda)R(\lambda)$. As reflectance values range between 0 and 1, a reflection corresponds to a filtering operation.

The sensation of the color perceived by the eye is modeled by tristimulus values found by evaluating the following definite integrals:

$$A = K \int_{\lambda_{inf}}^{\lambda_{sup}} S_r(\lambda)\bar{a}(\lambda)\, d\lambda \text{ with } A \in \{X, Y, Z\} \text{ and } a \in \{x, y, z\} \quad (1)$$

where K is a normalization coefficient, λ_{inf} and λ_{sup} are the bounds of the visible wavelength domain and $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ are the color matching functions of the XYZ colorimetric system defined by the "Commission Internationale de l'Eclairage" in 1931 [WS82].

In this paper, we shall use only one light source which will be assigned the value 1. Thus, XYZ tristimulus values are given by formula (1) with:

$$K = \frac{1}{\int_{\lambda_{inf}}^{\lambda_{sup}} S(\lambda)\bar{y}(\lambda)\, d\lambda}$$

The remainder of this paper is organized as follows. Section 2 presents known solutions dealing with full spectral information. Our method is introduced in section 3, some tests about this method are discussed in section 4 and a conclusion and some further developments are given in section 5.

2 Known solutions

Several techniques have been proposed that use spectral power distributions. They can be divided into two classes: Sampling methods and methods using a projection on a set of basis functions.

2.1 Sampling methods

2.1.1 Direct method (weighted ordinate). This method, the most common one, samples the visible spectrum every one, five or ten nanometers, and then uses a Riemann type integration. For our implementation, spectra on interval $[\lambda_{inf}, \lambda_{sup}] = [380nm, 700nm]$ were sampled every 5 nm, which gives 64 sample points. In the case of a spectrum with emission peaks (such as a fluorescent light), the power of each peak is spread on the 5 nm widthband which surrounds it.

Therefore the formulas used are:

$$A = K \sum_{i=1}^{i=64} S_r(\lambda_i)\bar{a}(\lambda_i) \text{ with } A \in \{X, Y, Z\} \text{ and } a \in \{x, y, z\}$$

.

This method gives good results but is very time and memory consuming. In the sequel, we shall consider this type of method as supplying the reference tristimulus values for our tests.

2.1.2 Gaussian quadrature method. The mathematical principle of the method is as follows: Let f be a function defined on interval [a,b] and w a weighting function positive on [a,b]. Then: $\int_a^b f(x)w(x)\,dx \approx \sum_{i=0}^{n} H_i f(x_i)$. For a definition of values H_i and x_i, $1 \leq i \leq n$, the reader may refer to [Wal75] or [Mey88].

In relation to our problem, this mathematical method has been used in the XYZ color space [Wal75] or in the AC_1C_2 color space [Mey88].

2.1.3 Segmentation of the visible wavelength domain. The main objective of this method [DMCP94] is to take all the types of spectra into account, whether they are continuous or not. Similarly to the Gaussian quadrature, this method may be used in the XYZ space or in the AC_1C_2 one.

In this method, the visible field of each light source in the scene is partitioned into a set of intervals by extracting first the non-zero portions of the spectrum and by isolating peaks from the continuum through a gradient calculation.

2.2 Projection on a set of basis functions

2.2.1 Use of polynomial functions. Raso and Fournier [RF91] suggest to use piecewise cubic polynomials to represent spectral power distributions. Interreflection computations are thus reduced to polynomial multiplications. To avoid obtaining very high degree for the final polynomial, the degree of the polynomials is kept constant by using a degree reduction technique based on Chebyshev polynomials. Finally color matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ are also represented by cubic piecewise polynomials to obtain XYZ tristimulus values.

2.2.2 Functions obtained by a characteristic vector analysis. The main advantage of the method presented in [Pee93] is to take the scene to be rendered into account, which should allow to adapt correctly to the future data to be treated.

The principle on which the method works is the following: From a family of spectral power distributions known to be representative of the scene, a characteristic vector analysis is performed providing m vectors $E_i(\lambda)$. These vectors $E_i(\lambda)$ are the eigenvectors associated to the m greatest eigenvalues of the covariance matrix for the inital vectors. The choice of a representative family may be solved in the following way: If the scene is made up of a light source S and n materials with reflectance $R_1, \ldots, R_n$, then spectra S, SR_i, $1 \leq i \leq n$, SR_iR_j, $1 \leq i, j \leq n, \ldots$ may be taken into account.

After this pre-processing step, each spectral power distribution is represented by its m coordinates in the basis: $(E_1(\lambda), \ldots, E_m(\lambda))$, and reflectance $R(\lambda)$ takes the form of an $m \times m$ matrix (R_{ij}), with:

$$R_{ij} = \int_{\lambda_{inf}}^{\lambda_{sup}} R(\lambda) E_i(\lambda) E_j(\lambda)\, d\lambda$$

Finally, for the integration step, a three lines and m columns matrix T has to be precomputed, with: $T(k,i) = \int_{\lambda_{inf}}^{\lambda_{sup}} E_i(\lambda)\bar{a_k}(\lambda)\, d\lambda$, with $\bar{a_1} = \bar{x}$, $\bar{a_2} = \bar{y}$ and $\bar{a_3} = \bar{z}$ and hence this method is entirely linear.

3 The adaptive solution

All the methods described in the previous section are efficient from a global time-saving point of view. But none of them ensures to give a permissible level of accuracy in every case. For example, we cannot answer the following question: How many sample points or basis functions are required to always obtain non perceptible errors?

We now suggest an adaptive algorithm for full spectral rendering. Currently, the error is computed in the XYZ color space, even if this space is known not to be uniform. The user first specifies an error interval $[\epsilon_{min}, \epsilon_{max}]$. A pre-processing step allows to create a data base where the light source spectrum, the reflectances of materials and the color matching functions are represented by binary trees. Then, during the processing step, the error produced is controlled: If the error is outside the given error interval, the representation of the spectral information is modified to better fit the user specifications. Finally, when the XYZ tristimulus values are required, they may be computed very quickly and without inaccuracy errors.

Let us specify the framework in which this algorithm is presented. It is mainly devoted to reflectance computations. Hence, the integration step is shortly tackled since it only appears as a consequence of previous computations. This study will be restricted to the case of a scene with only one type of light source. Finally, the problem of gathering the spectral information for each pixel of the image will not be discussed in this paper.

Actually, this method will take its whole significance when it is integrated in a progressive rendering algorithm such as a progressive ray tracer [MCP92] or a hierarchical radiosity [HSA91].

3.1 Pre-processing step

In order to obtain a progressive running of the algorithm, we suggest to project the spectral information on a set of basis functions, the main property of which being to allow a refinement of the representation as soon as an increasing number of terms is taken into account.

3.1.1 Choice of the set of basis functions. The spectral power distribution of light sources and reflectance surfaces are known every 5 nm, so these functions are piecewise constant. We chose to use scaling functions ϕ_i^j which are usually associated to adaptive Haar wavelets ψ_i^j. These scaling functions are defined by: $\phi_i^j(x) = 1\ \forall x \in [\lambda_{inf} + i\frac{\lambda_{sup}-\lambda_{inf}}{2^j}, \lambda_{inf} + (i+1)\frac{\lambda_{sup}-\lambda_{inf}}{2^j}]$, $\phi_i^j(x) = 0$ on the rest of interval $[\lambda_{inf}, \lambda_{sup}]$.

Actually, any linear combination of scaling functions such that:

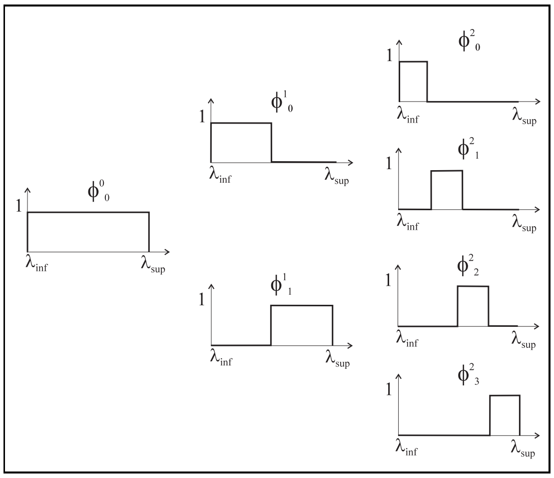

Fig. 1. Scaling functions

- the bounds of the visible wavelength domain are reached;
- no two functions overlap;
- to every ϕ_i^j is associated a coefficient f_{avi}^j which represents the average value of f on the interval inside which ϕ_i^j is non null ;

may be regarded as an approximation of function f defined on $[\lambda_{inf}, \lambda_{sup}]$.

Let us notice that this set of functions is stable under multiplication, which will be very significant for the computation of a spectrum-reflectance product. For every couple (f, g) of functions and for every triple (i, j, k) of indices:

$(f_{avi}^j \phi_i^j)(g_{avi}^j \phi_i^j) = (f_{avi}^j g_{avi}^j)\phi_i^j$ and $(f_{avi}^j \phi_i^j)(g_{avk}^j \phi_k^j) = 0$ if $i \neq k$.

3.1.2 Creation of the data base. Let us notice that, in this paragraph, function f may refer either to the light source spectrum or to any reflectance function.

In this stage, we compute for $0 \leq j \leq 6$ and $0 \leq i \leq 2^j - 1$ average value $f_{av[\lambda_i^j, \lambda_{i+1}^j]}$ on interval $[\lambda_i^j, \lambda_{i+1}^j]$ (where $\lambda_i^j = \lambda_{inf} + i\frac{\lambda_{sup} - \lambda_{inf}}{2^j}$) which minimizes $\sum_{\bar{a}=\bar{x},\bar{y},\bar{z}} (\epsilon_{f\bar{a}i}^j)^2$, where

$$\epsilon_{f\bar{a}i}^j = K \int_{\lambda_i^j}^{\lambda_{i+1}^j} (f(\lambda) - f_{av[\lambda_i^j, \lambda_{i+1}^j]})\bar{a}(\lambda)\, d\lambda \tag{2}$$

If we denote $f_{avi}^j = f_{av[\lambda_i^j, \lambda_{i+1}^j]}$, average value f_{avi}^j is given by formula:

$$f_{avi}^j = \frac{\sum_{\bar{a}=\bar{x},\bar{y},\bar{z}} (\int_{\lambda_i^j}^{\lambda_{i+1}^j} \bar{a}(\lambda)\, d\lambda \int_{\lambda_i^j}^{\lambda_{i+1}^j} f(\lambda)\bar{a}(\lambda)\, d\lambda)}{\sum_{\bar{a}=\bar{x},\bar{y},\bar{z}} (\int_{\lambda_i^j}^{\lambda_{i+1}^j} \bar{a}(\lambda)\, d\lambda)^2} \tag{3}$$

Of course, for $j = 6$ and $0 \leq i \leq 63$, we find exact values of piecewise constant functions S and R.

At the end of this stage, the light source spectrum and the reflectances of materials are represented by binary trees, the nodes of which store four fields:

- the average value $f^j_{av\,i}$ of the function on interval $[\lambda^j_i, \lambda^j_{i+1}]$;
- the three errors $\epsilon_{f\bar{x}}$, $\epsilon_{f\bar{y}}$ and $\epsilon_{f\bar{z}}$ associated to this average value.

We also compute binary trees for the color matching functions $\bar{x}$, $\bar{y}$ and $\bar{z}$. For each node, we must compute the following definite integrals:

$$\bar{a}^j_{av\,i} = K \int_{\lambda^j_i}^{\lambda^j_{i+1}} \bar{a}(\lambda)\, d\lambda$$

These binary trees will be used for the integration stage.

3.1.3 Representation of the light source spectrum. The purpose of this step is to obtain for the light source spectrum a representation as simple as possible, such that the error related to the reference tristimulus is less than ϵ_{max} in the XYZ color space. To do this, we carry out a traversal of the binary tree associated to the light source spectrum.

For the root of the tree, the error in the XYZ color space between the full spectrum and its representation by a single average value $S^0_{av\,0}$ on the visible domain is: $\epsilon_t = \sqrt{(\epsilon^0_{S\bar{x}0})^2 + (\epsilon^0_{S\bar{y}0})^2 + (\epsilon^0_{S\bar{z}0})^2}$.

Remark : Actually, square roots are not used in our current implementation.

If ϵ_t is greater than ϵ_{max}, this representation is not accurate enough. We must move one level down in the tree. The light source spectrum is thus approximated by two average values: $S^1_{av\,0}$ on interval $[\lambda_{inf}, (\lambda_{inf}+\lambda_{sup})/2[$ and $S^1_{av\,1}$ on interval $[(\lambda_{inf} + \lambda_{sup})/2, \lambda_{sup}]$. The new error made is:
$\epsilon_t = \sqrt{(\epsilon^1_{S\bar{x}0} + \epsilon^1_{S\bar{x}1})^2 + (\epsilon^1_{S\bar{y}0} + \epsilon^1_{S\bar{y}1})^2 + (\epsilon^1_{S\bar{z}0} + \epsilon^1_{S\bar{z}1})^2}$.

If ϵ_t is again greater than ϵ_{max}, the representation of the spectrum must be refined. But this time, we look for the half-interval for which the error made is the greater, that is the half-interval for which $\sqrt{(\epsilon^j_{S\bar{x}i})^2 + (\epsilon^j_{S\bar{y}i})^2 + (\epsilon^j_{S\bar{z}i})^2}$ is maximal. Let us suppose we have found the first half-interval. The light source spectrum is then approximated by $S^2_{av\,0}$ and $S^2_{av\,1}$ on the first two intervals with length a quarter of the visible domain, and by $S^1_{av\,1}$ on the rest. The new value of the error is:
$\epsilon_t = \sqrt{(\epsilon^2_{S\bar{x}0} + \epsilon^2_{S\bar{x}1} + \epsilon^1_{S\bar{x}1})^2 + (\epsilon^2_{S\bar{y}0} + \epsilon^2_{S\bar{y}1} + \epsilon^1_{S\bar{y}1})^2 + (\epsilon^2_{S\bar{z}0} + \epsilon^2_{S\bar{z}1} + \epsilon^1_{S\bar{z}1})^2}$.

This process is repeated until the error becomes less than ϵ_{max}. At the end of this stage, the visible domain is cut in a series of intervals.

3.2 Computation of a spectrum-reflectance product

We now manage the interaction of light with surface materials. Let us consider an incident spectrum, called S_r, which is not necessarily the light source spectrum. It is defined as a series of intervals partitioning the visible wavelength domain. An average value S^j_{ravi} and three errors $\epsilon^j_{S_r\bar{x}i}$, $\epsilon^j_{S_r\bar{y}i}$ and $\epsilon^j_{S_r\bar{z}i}$ are associated with each interval.

Now, let us call R the reflectance of a given material and let us consider the same partition for R than that for the incident spectrum. By looking at the binary tree associated to R, we can find corresponding values R^j_{avi} and errors $\epsilon^j_{R\bar{x}i}$, $\epsilon^j_{R\bar{y}i}$ and $\epsilon^j_{R\bar{z}i}$.

For each interval of the current partition, the value of the reflected spectrum S_rR is $(S_rR)^j_{avi} = S^j_{ravi} * R^j_{avi}$.

We now have to estimate the new errors. By definition:

$$\epsilon^j_{(S_rR)\bar{a}i} = K \int_{\lambda^j_i}^{\lambda^j_{i+1}} (S_r(\lambda)R(\lambda) - S^j_{ravi}R^j_{avi})\bar{a}(\lambda)\,d\lambda$$

but

$$\epsilon^j_{(S_rR)\bar{a}i} = K \int_{\lambda^j_i}^{\lambda^j_{i+1}} (S_r(\lambda)R(\lambda) - S^j_{ravi}R(\lambda) + S^j_{ravi}R(\lambda) - S^j_{ravi}R^j_{avi})\bar{a}(\lambda)\,d\lambda$$

$$= K \int_{\lambda^j_i}^{\lambda^j_{i+1}} (S_r(\lambda) - S^j_{ravi})R(\lambda)\bar{a}(\lambda)\,d\lambda + K \int_{\lambda^j_i}^{\lambda^j_{i+1}} S^j_{ravi}(R(\lambda) - R^j_{avi})\bar{a}(\lambda)\,d\lambda$$

$$= R^j_{avi}K \int_{\lambda^j_i}^{\lambda^j_{i+1}} (S_r(\lambda) - S^j_{ravi})\bar{a}(\lambda)\,d\lambda + S^j_{ravi}K \int_{\lambda^j_i}^{\lambda^j_{i+1}} (R(\lambda) - R^j_{avi})\bar{a}(\lambda)\,d\lambda$$

$$+K \int_{\lambda^j_i}^{\lambda^j_{i+1}} (S_r(\lambda) - S^j_{ravi})(R(\lambda) - R^j_{avi})\bar{a}(\lambda)\,d\lambda$$

If we assume residual term $K\int_{\lambda^j_i}^{\lambda^j_{i+1}}(S_r(\lambda)-S^j_{ravi})(R(\lambda)-R^j_{avi})\bar{a}(\lambda)\,d\lambda$ negligible, then $\epsilon^j_{(S_rR)\bar{a}i} \approx R^j_{avi}\epsilon^j_{S_r\bar{a}i} + S^j_{ravi}\epsilon^j_{R\bar{a}i}$.

As a result of the assumption made, we only obtain an estimation of the error. We shall see later, in section (4), if this estimation is enough accurate to take the right decision for the treatment of the data.

After the computation of this spectrum-reflectance product, we must estimate the total error made for spectrum S_rR, in relation to the reference calculation. We have :

$$\epsilon_t = \sqrt{\Big(\sum_{intervals} \epsilon^j_{S_rR\bar{x}i}\Big)^2 + \Big(\sum_{intervals} \epsilon^j_{S_rR\bar{y}i}\Big)^2 + \Big(\sum_{intervals} \epsilon^j_{S_rR\bar{z}i}\Big)^2}$$

Three cases may occur:

- ϵ_t belongs to interval $[\epsilon_{min}, \epsilon_{max}]$. The representation of the reflected spectrum is satisfactory. We may either carry out a new spectrum-reflectance product, or a transformation to XYZ tristimulus values by integration;

- $\epsilon_t > \epsilon_{max}$. We must refine the representation of the spectrum. For that, we must detect the interval for which $\sqrt{(\epsilon^j_{S_r\bar{x}i})^2 + (\epsilon^j_{S_r\bar{y}i})^2 + (\epsilon^j_{S_r\bar{z}i})^2}$ is maximal. We must find the list of successive reflections leading to the current result. Then, in the binary tree of the light source spectrum and in the binary trees linked to the reflectances appearing in the list, we must consider the two children of $node^j_i$ corresponding to the designated interval. For these two sub-intervals, we must perform spectrum-reflectance products as described above. We thus obtain two new average values, and two error triples $\epsilon_{S_r\bar{x}}$, $\epsilon_{S_r\bar{y}}$ and $\epsilon_{S_r\bar{z}}$ associated with these approximations. After this refinement test, the total error is estimated in order to determine in which case we are;

- $\epsilon_t < \epsilon_{min}$. The representation of the spectrum is too sharp for our needs. The idea is to avoid unnecessary rendering calculations in relation to psycho-visual appraisement of the result. We are going to merge two intervals in only one. In the current list of intervals $[\lambda^j_i, \lambda^j_{i+1}]$, let us look for those having the most important depth (j maximum). As a result of our algorithm, these intervals always go in pairs. We choose the pair such that $(S_rR^{j_{max}}_{avi} - S_rR^{j_{max}}_{avi+1})$ is minimum. Let us call i_{min} the coordinate i of the first interval. We replace this pair of intervals by a new interval, the coordinates of which are $i_{new} = i_{min}/2$ and $j_{new} = j_{max} - 1$ and we take: $S_rR^{j_{new}}_{avi_{new}} = \frac{S_rR^{j_{max}}_{avi_{min}} + S_rR^{j_{max}}_{avi_{min}+1}}{2}$. Finally, the new errors associated with each color matching function $\bar{a}(\lambda)$ will be:

$$\epsilon^{j_{new}}_{S_rR\bar{a}i_{new}} = \epsilon^{j_{max}}_{S_rR\bar{a}i_{min}} + \epsilon^{j_{max}}_{S_rR\bar{a}i_{min}+1} +$$
$$\left(\int_{\lambda^{j_{max}}_{i_{min}}}^{\lambda^{j_{max}}_{i_{min}+1}} \bar{a}(\lambda)\, d\lambda\right)(S_rR^{j_{max}}_{avi_{min}} - S_rR^{j_{new}}_{avi_{new}}) +$$
$$\left(\int_{\lambda^{j_{max}}_{i_{min}+1}}^{\lambda^{j_{max}}_{i_{min}+2}} \bar{a}(\lambda)\, d\lambda\right)(S_rR^{j_{max}}_{avi_{min}+1} - S_rR^{j_{new}}_{avi_{new}})$$

  At the end of this merge step, the total error is re-estimated in order to determine in which case we are.

3.3 Change to XYZ tristimulus values

At the end of the spectral rendering stage, a spectrum $S_r(\lambda)$ computed at each pixel appears in the form of a list of intervals. An average value S^j_{ravi} is linked to each interval. The change to XYZ tristimulus values comes down to an integration of the spectrum weighted by the color matching functions. As the spectrum is a piecewise constant function, this integration is easily performed by multiplying S^j_{ravi} by the value of the color matching function on the corresponding interval, which has been computed during the pre-processing step.

The XYZ tristimulus values are thus obtained as: $A = \sum_{intervals} S^j_{ravi}\bar{a}^j_{avi}$

3.4 Outline of the algorithm

To sum up, the adaptive representation algorithm of spectral data may be written as follows:

```
/* We suppose the data base built as binary trees related to the light source spectrum,
the reflectances and the color matching functions */
Initialization (Spectrum_Binary_Tree, S);
/* S is a list of intervals containing the following information: S_av, ε_x, ε_y, ε_z */
Create_Empty (List_of_Successive_Reflection);
/* This list is needed for Refinement computation */
S_r = S;
While ( radiance < threshold)
/* this test could be replaced by a psycho-visual test focused on luminous radiance */
{
    Spectrum_Reflectance_Product (S_r, Reflectance_Binary_Tree,S_r R);
    Add(Reflectance_Binary_Tree, List_of_Successive_Reflection);
    Computation_Total_Error (S_r R,ε_t);
    While ((ε_t > ε_max) or (ε_t < ε_min))
    {
        If (ε_t > ε_max) Then Refinement (S_r R,List_of_Successive_Reflection);
            Else Merge (S_r R);
        Updating_Total_Error (S_r R,ε_t);
    }
    S_r = S_r R;
}
Computation_XYZ_Coordinates (S_r,XYZ_Color);
```

4 Tests

4.1 Test procedure

For our tests, we used three types of light sources: CIE Standard Illuminants C and D 6500 with continuous spectra, and CIE representative fluorescent illuminant F2 with four emission lines (405 nm, 435 nm, 545 nm and 580 nm) ([Hun95]), twenty-four reflectances taken from the Macbeth Color Checker chart ([CMD76]) and the CIE *Luv* color space to estimate the perceptibility of the errors made. The tests were made on the light source spectra, primary reflections (SR_i, $1 \leq i \leq 24$), secondary reflections (SR_iR_j, $1 \leq i,j \leq 24$) and tertiary reflections ($SR_iR_jR_k$, $1 \leq i,j,k \leq 24$).

In each case, we computed the average number of required intervals and we detailed the different operations used. We also counted the number of measure errors, *ie* the number of cases where it was wrongly estimated that the error was greater than ϵ_{max} (this case may result from the approximation made for the calculation of the error after a spectrum-reflectance product, see section 3.2).

All the results are contained in Table 1, with $\epsilon_{max} = 2.10^{-3}$ and $\epsilon_{min} = 5.10^{-4}$.

Source	C	D6500	F2
Refinement of S : Number of intervals	8	7	11
XYZ error	0.0014	0.0013	0.0015
Luv error	0.26	0.33	0.27
Primary reflections			
Average number of intervals	8.42	7.83	10.17
Standard deviation of the number of intervals	1.63	1.79	1.91
Average Luv error	0.91	0.88	0.84
Maximum Luv error	2.34	1.69	1.87
Standard deviation of the Luv error	0.53	0.41	0.49
OPERATIONS			
None	41.70 %	25.00 %	50.00 %
Refinement	45.80 %	58.33 %	12.50 %
Merge	12.50 %	16.67 %	37.50 %
Fluctuation	0.00 %	0.00 %	0.00 %
Measure error	29.17 %	29.17 %	41.67 %
Secondary reflections			
Average number of intervals	7.15	6.91	7.33
Standard deviation of the number of intervals	2.03	2.00	2.66
Average Luv error	1.31	1.33	1.22
Maximum Luv error	5.95	6.62	6.27
Standard deviation of the Luv error	0.84	0.93	0.95
OPERATIONS			
None	46.35 %	47.22 %	27.78 %
Refinement	4.51 %	9.72 %	0.87 %
Merge	46.70 %	39.76 %	67.71 %
Fluctuation	2.43 %	3.30 %	3.65 %
Measure errors	11.63 %	10.76 %	13.54 %
Tertiary reflections			
Average number of intervals	5.25	5.16	4.96
Standard deviation of the number of intervals	1.94	1.88	1.99
Average Luv error	1.53	1.57	1.51
Maximum Luv error	8.07	8.12	10.79
Standard deviation of the Luv error	1.07	1.15	1.28
OPERATIONS			
None	26.92 %	29.72 %	22.08 %
Refinement	0.52 %	1.08 %	0.05 %
Merge	68.81 %	65.14 %	73.21 %
Fluctuation	3.75 %	4.06 %	4.67 %
Measure error	6.27 %	6.00 %	7.21 %

Table 1

Remark: "None" means that ϵ_t was directly found inside the right interval. "Fluctuation" shows a case where after $\epsilon_t > \epsilon_{max}$, we obtain $\epsilon_t < \epsilon_{min}$ or conversely.

4.2 Analysis of the results

The cases where no correction (merge or refinement) is needed are not the most numerous. However, when the number of interreflections increases, the average number of intervals decreases. In fact, more and more merges and less and less refinements are made when the number of interreflections raises. This may be explained by the fact that the initial light source spectrum is filtered at each reflection, and therefore has a weaker average level of energy, and especially a flattter and flatter profile.

It may also be seen that perceptual errors are increasing with the number of reflections. To check that the error belongs to some interval $[\epsilon_{min}, \epsilon_{max}]$ of the XYZ color space is not sufficient. Let us notice that the bounds of the interval taken for out tests were chosen empirically. They seemed to be a good compromise to keep an average perceptual error in the Luv space around 1, which is the threshold for non perceptible errors.

Measure errors are rather good. In spite of the simplifying assumption made, the estimation mode used allows to take, in general, the right decision. Only few fluctuations appear. This is ensured by a ratio between ϵ_{max} and ϵ_{min} around 4. Finally, the average number of intervals used allows to make a change to XYZ tristimulus values around 10 times faster.

To take this last remark into account, a final test was performed. Running times for the reference computation and for the progressive algorithm were calculated. The whole set of tertiary interreflections (13 824 cases) was computed without taking fluctuations into account and changed to XYZ tristimulus values. The results are as follows:

	Reference computation	Adaptive computation
Pre-processing	no	yes: 30 ms
Spectrum-reflectance product	580 ms	590 ms
Change to XYZ coordinates	240 ms	30 ms
Total time	820 ms	620 ms

Table 2

5 Conclusion and further developments

We think the algorithm described in this paper is interesting in more than one account. To try to control colorimetric errors during a rendering calculation seems to be a research theme as important as those which consist in saving time or improving the realism of the pictures.

However, our method must be improved. First, the controlled error is not of a perceptual type. Secondly, we would like to be able to handle several types of light sources in a given scene. Finally, we are trying to obtain an upper bound for the residual term neglected during the computation of the error $\epsilon^{j}_{(S_r R)\bar{a}i}$.It's a work in progress.

We are also currently including this algorithm in a spectral ray tracer. The purpose of this integration is to use around 10 times less rays than with an even sampling of the visible wavelength domain. This should allow to obtain a very efficient spectral ray tracer.

References

[CMD76] C.S. Mc Camy, H. Marcus, and J.G. Davidson, *A color rendition chart*, Journal of Applied Photographics Engineering **2** (1976), no. 3, 95–99.

[DMCP94] P. Deville, S. Merzouk, D. Cazier, and J.C. Paul, *Spectral data modeling for a lighting application*, Eurographics 94 **13** (1994), no. 3, 97–106.

[HSA91] P. Hanrahan, D. Salzman, and L. Aupperle, *A rapid hierarchical radiosity algorithm*, Computer Graphics **25** (1991), no. 4, 197–206.

[Hun95] R.W.G. Hunt, *Measuring colour*, Ellis Horwood, 1995.

[Mai96] Jean-Luc Maillot, *Pseudo-réalisme et progressivité pour le tracé de rayons*, Ph.D. thesis, Ecole des Mines de Saint-Etienne, France, September 1996.

[MCP92] Jean-Luc Maillot, Laurent Carraro, and Bernard Peroche, *Progressive ray tracing*, Third Eurographics Workshop on Rendering, Bristol, England (1992), 9–19.

[Mey88] G. Meyer, *Wavelength selection for synthetic image generation*, Computer Vision, Graphics and Image Processing (1988), no. 41, 57–79.

[Pee93] S. Peercy, *Linear color representation for full spectral rendering*, Computer Graphics **27** (1993), 191–198.

[RF91] M. G. Raso and A. Fournier, *A piecewise polynomial approach to shading using spectral distribution*, Proceedings of Graphics Interface 91, Calgary, Canada (1991), 40–46.

[Roe93] Marc Roelens, *Un environnement pour le tracé de rayons utilisant une modélisation par arbre de construction*, Ph.D. thesis, Ecole des Mines de Saint-Etienne, France, February 1993.

[Wal75] R. Wallis, *Fast computation of tristimulus values by use of gaussian quadrature*, J. Opt. Soc. Am **65** (1975), no. 1, 91–94.

[WS82] G. Wysecki and W. Stiles, *Color science: Concepts and methods, quantitative data and formulae*, John Wiley and Sons, New York, 1982.