Un lancer de rayons spectral adaptatif

Jean Claude Iehl et 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

Résumé : *Nous proposons, dans cet article, une méthode de rendu spectral basée sur un algorithme de type lancer de rayons, dont l'objectif est un contrôle perceptuel de l'erreur commise. Pour cela, nous utilisons une représentation adaptative des données spectrales des sources et des matériaux, conduisant, pour chaque pixel, à l'évaluation d'une expression. Cet article présente un état d'avancement de notre projet.*

Mots-clés : Rendu spectral, lancer de rayons, représentation adaptative, contrôle de l'erreur

1 Introduction

Les algorithmes de rendu basés sur les lois de la physique ont pour objectif de générer des images réalistes, en simulant le plus efficacement possible le comportement de la lumière et ses interactions avec les matériaux de la scène. Un des élements importants pour cette simulation est l'aspect spectral de la lumière, que la représentation des couleurs à l'aide de trois composantes ne permet pas de prendre vraiment en compte.

Ce problème est perçu comme important depuis plus de dix ans maintenant, et un certain nombre de solutions ont déja été proposées pour représenter des données spectrales. Une partie de ces solutions reposent sur des techniques d'échantillonnage plus ou moins sophistiquées [Mey88] [DMCP94] [ZCB97]. D'autres utilisent une projection sur un espace de fonctions de base [RF91] [Pee93] [RP97].

Notre travail, qui se situe dans le prolongement de [RP97], a pour but d'élaborer un logiciel de rendu prenant en compte les représentations spectrales des sources de lumières et des matériaux. Un des points clés de la spécification de notre logiciel est que nous voulons contrôler l'erreur perceptuelle commise en chaque pixel. L'idée est qu'il est en effet inutile de faire des calculs complexes qui fournissent des résultats que l'observateur n'est pas capable de différencier d'autres beaucoup plus simples. Cette approche n'est pas véritablement nouvelle en synthèse d'images et des travaux ont déjà été publiés, tant en lancer de rayons [BM95] [BM98], qu'en radiosité [GH97] [Mys98].

Pour notre part, nous avons choisi de travailler dans un environnement de type lancer de rayons en basant les calculs sur une représentation adaptative des spectres dirigée par une évaluation de l'erreur commise.

Cet article sera organisé ainsi : dans le paragraphe 2, nous effectuerons quelques rappels sur le rendu spectral. Nous décrirons dans le paragraphe 3 la représentation adaptative choisie, qui est basée sur celle décrite dans [RP97]. Le paragraphe 4 sera consacré à la mise en œuvre dans un lancer de rayons. Des résultats seront présentés dans le paragraphe 5, une conclusion et des perspectives dans le paragraphe 6.

2 Rendu spectral

La couleur d'un point de la scène est la réponse du système visuel à l'énergie provenant de ce point (cf. figure 1). Cette énergie est le résultat des interactions de la lumière avec les matériaux des objets de la scène.

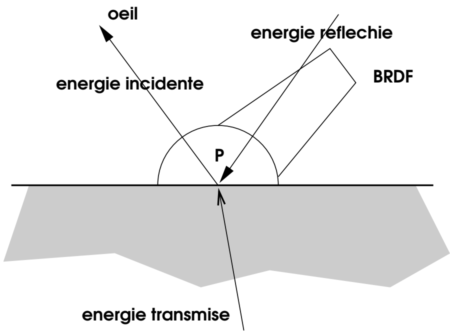

FIG. 1 – *couleur d'un point image*

Ces interactions se calculent classiquement en échantillonnant régulièrement le domaine visible. Cette solution peut difficilement être considérée comme acceptable, puisqu'elle revient à calculer une image par longueur d'onde échantillonnée, soit plus d'une soixantaine pour une précision couleur correcte en échantillonnant tous les $5nm$ [Hun91]. On pourrait évidemment réduire le nombre d'échantillons et utiliser un échantillonnage plus "intelligent" [Mey88] [DMCP94] [ZCB97] mais ce n'est pas une solution satisfaisante : en raisonnant globalement, on perd la possibilité d'un contrôle plus ou moins local de l'erreur perceptuelle commise. Une solution locale qui s'adapte au comportement du matériau et à l'énergie présente permettrait de contrôler l'erreur et de garantir le résultat final. Il faut donc choisir une représentation des spectres qui permette de simuler les interactions en utilisant plus ou moins d'échantillons. Pour une efficacité maximale, il serait également souhaitable que les calculs se fassent directement sur la représentation et pas sur chaque échantillon, l'idée étant d'effectuer les calculs pour un groupe d'ondes [RP97] [Pee93]. Ce sont ces constatations qui nous ont permis de choisir une représentation adaptative.

3 Une représentation adaptative

Notre choix s'est porté sur une décomposition sur une base de fonctions d'échelles [RP97] : les ondelettes de Haar.

$$\begin{aligned} f_i^j(x) &= 1 \quad \text{si} \quad x \in \left[\frac{i}{2^j},\ \frac{i+1}{2^j}\right] \\ &\ 0 \quad \text{sinon} \end{aligned}$$

Ces fonctions autorisent une décomposition hiérarchique naturelle et sont stables par multiplication, ce qui permettra de définir la représentation du résultat d'un produit à partir des représentations de ses opérandes.

3.1 Représentation d'un spectre

Les spectres sont décomposés sur une base de fonctions associée au domaine visible. La subdivision est dichotomique : la racine de la décomposition est associée au domaine complet $[\lambda_{inf},\ \lambda_{sup}]$. Chaque coefficient f_i^j de la décomposition correspond donc à un intervalle dont les bornes sont définies par : $\lambda_a = \lambda_{inf} + i\frac{\lambda_{sup}-\lambda_{inf}}{2^j}$, $\lambda_b = \lambda_{inf} + (i+1)\frac{\lambda_{sup}-\lambda_{inf}}{2^j}$. Le domaine visible est donc représenté par un arbre binaire décrivant cette décomposition (cf. figure 2).

3.1.1 Représentation sur un intervalle

Le coefficient associé à un intervalle représente une valeur moyenne. L'objectif étant de garantir l'erreur couleur associée à l'énergie incidente, on choisit une valeur f_i^j minimisant l'erreur

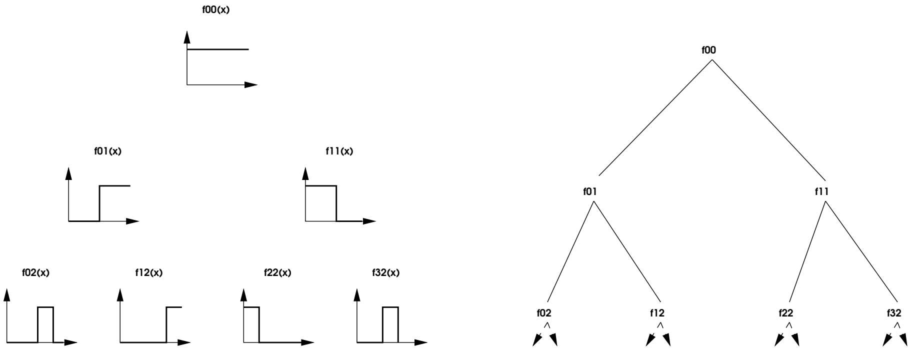

FIG. 2 – *découpage du domaine visible et arbre de décomposition*

couleur ϵ_i^j sur l'intervalle $[\lambda_a \ \lambda_b]$:

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

où $\bar{x}(\lambda), \bar{y}(\lambda)$ et $\bar{z}(\lambda)$ représentent les sensibilités couleurs de l'espace XYZ [WS82].

L'erreur couleur $\varepsilon_i^j = \sqrt{(\epsilon_{\bar{x}i}^j)^2 + (\epsilon_{\bar{y}i}^j)^2 + (\epsilon_{\bar{z}i}^j)^2}$ associée à l'intervalle est l'intégration de la différence entre les échantillons $f(\lambda)$ et la valeur moyenne f_i^j pour chaque sensibilité $\bar{a} = \bar{x}, \bar{y}, \bar{z}$:

$$\epsilon_{\bar{a}i}^j = \int_{\lambda_a}^{\lambda_b} (f(\lambda) - f_i^j)\bar{a}(\lambda)\, d\lambda \tag{2}$$

Un intervalle est donc représenté par sa valeur moyenne f_i^j et les erreurs couleurs associées $\epsilon_{\bar{x}i}^j, \epsilon_{\bar{y}i}^j, \epsilon_{\bar{z}i}^j$.

3.1.2 Représentation du modèle local : BRDF spectrale

Le comportement local d'un matériau est décrit par une fonction, notée BRDF, qui exprime le rapport entre l'énergie incidente et l'énergie réfléchie pour une longueur d'onde donnée. Il faut donc étendre l'utilisation de cette fonction à tout le domaine visible. Pour pallier le manque de mesures de BRDF spectrales, il est usuel de considérer qu'elles sont séparables en un terme géométrique (analytique ou mesuré) et un terme spectral. La réflectance spectrale et le facteur de Fresnel sont disponibles pour un nombre important de matériaux. Les modèles de C. Schlick [Sch94] et G. Ward [War92] introduisent cette séparation. Nous avons utilisé le modèle défini dans [Sch94] (repère et notations figure 3) :

$$f_{r\lambda}(t, u, v, v') = S_\lambda(u) D(t, v, v')$$

où $D(t, v, v')$ est le terme géométrique et $S_\lambda(u) = C_\lambda + (1 - C_\lambda)(1 - u)^5$ le terme spectral.

Les coefficients C_λ définissent le comportement spectral du matériau. Pour une géométrie fixée, on peut réécrire $S_\lambda(u)$ sous la forme $S_u(\lambda) = C(\lambda) + (1 - C(\lambda))(1 - u)^5$. On peut ensuite facilement déduire une représentation adaptative de $S_u(\lambda)$ de la représentation des coefficients $C(\lambda)$.

3.2 Représentation d'une opération

Une opération est représentée par un arbre dont la racine précise l'opérateur et les sous-arbres les opérandes.

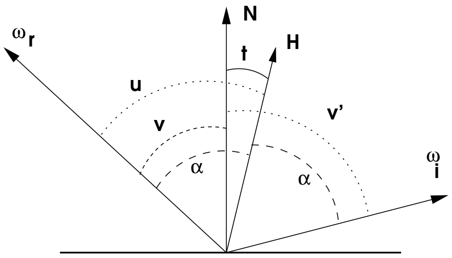

FIG. 3 – *repère local et notations*

On veut déterminer le résultat de l'opération sur un intervalle particulier en évitant de faire le calcul pour tous les échantillons. Il faut donc déterminer la représentation de l'intervalle résultat, c'est à dire sa valeur moyenne et l'erreur associée, à partir des représentations des intervalles des opérandes. On peut évidement déterminer la représentation de tous les intervalles de la décomposition. Dans un souci d'efficacité, seuls les intervalles réellement nécessaires seront calculés ; ils seront déterminés lors de l'évaluation adaptative.

3.3 Évaluation adaptative d'une opération

3.3.1 stratégie générale

L'évaluation d'une opération doit déterminer une représentation, c'est à dire un ensemble d'intervalles, assurant une erreur couleur inférieure à un seuil fixé. Afin de restreindre le nombre d'intervalles, on affine le découpage dans les zones produisant des erreurs importantes en subdivisant l'intervalle dont l'erreur est la plus importante. Il est assez naturel de commencer le calcul du résultat d'une opération par l'intervalle racine de la décomposition : le domaine visible complet. La construction de la représentation d'une opération est récursive : elle parcourt l'arbre de la décomposition du domaine visible. On peut résumer cette stratégie par l'algorithme:

```
évaluation d'une opération :
    calculer la représentation de la racine et l'insérer dans la liste de candidats
    calculer la couleur et l'erreur associée à cette représentation
    tant que l'erreur est supérieure à un seuil donné
        recupérer l'intervalle produisant l'erreur max dans la liste de candidats
        subdiviser cet intervalle :
            l'enlever de la représentation
            calculer la représentation de ses deux fils
            ajouter les deux fils à la représentation
            insérer les deux fils dans la liste de candidats
            mettre à jour la couleur et l'erreur de la représentation
    fin
```

Il ne reste plus qu'à préciser comment obtenir la représentation du résultat des différentes opérations nécessaires pour effectuer un calcul de rendu.

3.3.2 évaluation d'une somme

Il faut déterminer la valeur moyenne et l'erreur couleur d'une somme de deux spectres sur un intervalle. Soient $S1(\lambda)$, et $S2(\lambda)$ deux distributions d'énergie, $S1_i^j$, $S2_i^j$ leurs valeurs moyennes, $\epsilon^j_{S1\bar{a}i}$, $\epsilon^j_{S2\bar{a}i}$ leurs erreurs couleurs sur l'intervalle $[\lambda_a,\ \lambda_b]$. On veut déterminer une représentation de $S(\lambda) = S1(\lambda) + S2(\lambda)$. Les équations (1) et (2) étant linéaires par rapport à f, on peut représenter une somme par $S_i^j = S1_i^j + S2_i^j$ et $\epsilon^j_{S\bar{a}i} = \epsilon^j_{S1\bar{a}i} + \epsilon^j_{S2\bar{a}i}$.

3.3.3 évaluation d'un produit

De même, on veut déterminer la représentation de $S(\lambda) = S1(\lambda)S2(\lambda)$. L'intégration couleur s'écrit [WS82] :

$$X = \int_{\lambda_a}^{\lambda_b} (S1(\lambda)S2(\lambda))\bar{x}(\lambda)\, d\lambda$$

En introduisant les valeurs moyennes et après quelques remaniements :

$$X = \int_{\lambda_a}^{\lambda_b} (S1(\lambda) - S1_i^j)(S2(\lambda) - S2_i^j)\bar{x}(\lambda)\, d\lambda + S1_i^j \epsilon^j_{S2\bar{x}i} + S2_i^j \epsilon^j_{S1\bar{x}i} + S1_i^j S2_i^j \int_{\lambda_a}^{\lambda_b} \bar{x}(\lambda)\, d\lambda$$

Malgré de nombreuses tentatives, nous n'avons pu trouver d'estimation satisfaisante du premier terme de cette formule. Il est actuellement négligé, ce qui conduit, généralement, à sous-estimer l'erreur commise. La représentation d'un produit se fait donc par $S_i^j = S1_i^j S2_i^j$ et $\epsilon^j_{S\bar{a}i} = S1_i^j \epsilon^j_{S2\bar{a}i} + S2_i^j \epsilon^j_{S1\bar{a}i}$.

3.3.4 évaluation de la BRDF

On veut déterminer la représentation du terme spectral de la BRDF à partir de celle des coefficients. Pour une géométrie donnée, ce terme s'écrit : $S_u(\lambda) = C(\lambda) + (1 - C(\lambda))f_u$ où f_u est l'approximation du terme de Fresnel (voir [Sch94]). Soient C_i^j la valeur moyenne et $\epsilon^j_{C\bar{a}i}$ les erreurs couleurs associées à $C(\lambda)$. Après introduction des valeurs moyennes dans l'intégration couleur et simplification, on obtient une représentation par $S^j_{ui} = C_i^j(1 - f_u) + f_u$ et $\epsilon^j_{S_u i} = (1 - f_u)\epsilon^j_{C\bar{a}i}$.

3.3.5 évaluation d'un produit par un coefficient

D'après la linéarité des équations (1) et (2), la représentation de $S(\lambda) = a \cdot S1(\lambda)$ est : $S_i^j = a \cdot S1_i^j$ et $\epsilon^j_{Si} = a \cdot \epsilon^j_{S1i}$

3.4 Évaluation d'une expression

L'énergie réfléchie $L_\lambda(P, \vec{\omega_r})$ au point P de la scène et dans la direction $\vec{\omega_r}$ est le résultat de la propagation de la lumière et de ses interactions avec les matériaux rencontrés :

$$L_\lambda(P, \vec{\omega_r}) \;=\; L_{e\lambda}(P, \vec{\omega_r}) \;+\; \int_{\Omega_i} f_r(P, \vec{\omega_i} \to \vec{\omega_r})\, L_{i\lambda}(P, \vec{\omega_i})\, (\vec{N} \cdot \vec{\omega_i})\, d\omega_i$$

où $L_{e\lambda}(P, \vec{\omega_r})$ est la luminance émise par le point P dans la direction $\vec{\omega_r}$, $L_{i\lambda}(P, \vec{\omega_i})$ la luminance incidente au point P selon la direction $\vec{\omega_i}$ dans l'angle solide $d\omega_i$ et $f_r(P, \vec{\omega_i} \to \vec{\omega_r})$ la BRDF au point P pour le couple de directions $(\vec{\omega_i}, \vec{\omega_r})$. Pour une géométrie fixée, la BRDF se réduit à un terme spectral multiplié par un coefficient.

L'évaluation itérative oblige à construire explicitement l'expression de l'énergie incidente au point image. A cause de la formulation récursive de l'énergie incidente, le résultat d'une opération doit pouvoir être l'opérande d'une autre opération.

3.4.1 stratégie générale

La stratégie déterminée pour une opération s'applique aussi très bien à l'évaluation d'une expression. La représentation recherchée est la représentation du résultat de l'opération racine de l'expression. Il suffit de calculer la représentation de l'intervalle racine de chaque opération pour initialiser l'erreur de la représentation et ensuite subdiviser l'intervalle dont l'erreur est la

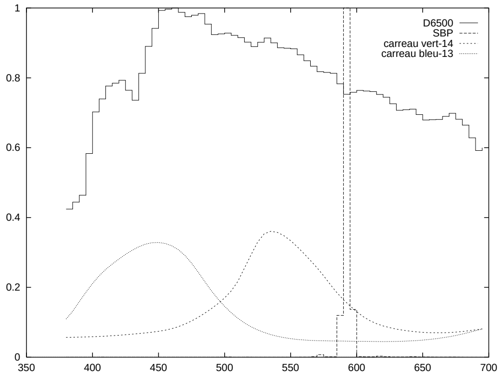

FIG. 4 – *données spectrales*

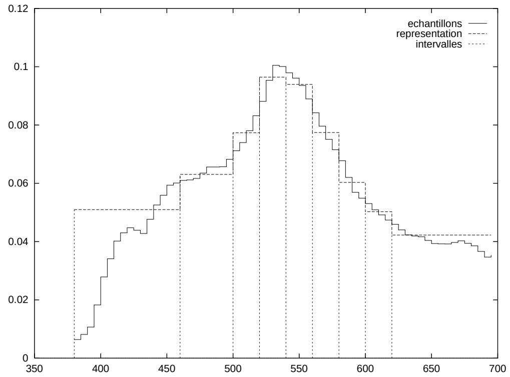

FIG. 5 – *spectre réfléchi du vert-14 éclairé par une source D6500: 9 intervalles, erreur CIELAB 0.68*

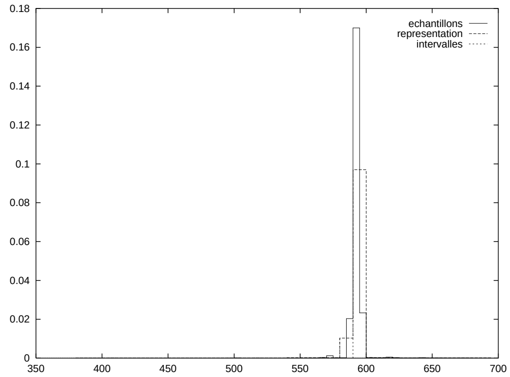

FIG. 6 – *spectre du bleu-13 éclairé par une source sodium: 6 intervalles, erreur CIELAB 1.64*

plus importante. Avec cette stratégie, tous les spectres sont découpés de la même manière, ce qui simplifie considérablement la définition des opérations ([DMCP94] et [ZCB97] utilisent des découpages différents pour chaque opérande).

```
évaluation d'une expression :
    calculer la racine de la représentation de chaque opération de l'expression
    calculer la couleur et l'erreur de cette représentation
    tant que l'erreur est supérieure à un seuil donné
        recupérer l'intervalle de la représentation dont l'erreur est la plus importante
        subdiviser cet intervalle :
            l'enlever de la représentation
            calculer la représentation de ses deux fils (parcours de l'expression)
            ajouter les deux fils à la représentation
            insérer les deux fils dans la liste de candidats
            mettre à jour la couleur et l'erreur de la représentation
    fin
```

3.4.2 Exemples

L'expression représentée figure 5 correspond à l'image réfléchie d'un mur (matériau construit avec la réflectance du carreau vert-14 du ColorChecker [MMD76] et une BRDF simple isotrope [Sch94], voir la figure 4 pour plus de détails) éclairé par une source surfacique de type D6500 [WS82]. L'expression est composée de 50 opérateurs. La représentation déterminée pour un seuil CIELAB de 3 [WS82] comporte 9 intervalles pour une erreur CIELAB de 0.68. La figure 6 correspond à la réflexion d'une source surfacique de type sodium basse pression (4 raies) sur un matériau construit avec la réflectance du carreau bleu-13 du ColorChecker et une BRDF simple isotrope. L'expression comporte 20 opérateurs et la représentation 6 intervalles pour une erreur CIELAB de 1.64.

Remarque : La stratégie de subdivision choisie s'arrête dès que l'erreur est inférieure au seuil. Dans les exemples des figures 5 et 6, la dernière subdivision a permis de diminuer considérablement l'erreur. Il existe certainement des découpages permettant d'obtenir une erreur plus proche du seuil avec moins d'intervalles. Ce comportement s'explique en remarquant que la stratégie de subdivision choisie est de type *best first* [CLR92]. On peut essayer d'autres techniques de recherche basées sur des algorithmes de planifications utilisant des heuristiques tels que A^* [CLR92] [Nil82] ou sa version dynamique D^* [Ste94] [Ste95]. Ces algorithmes nécessitent d'établir le graphe complet des subdivisions du domaine visible pour fonctionner. Dans le cas présent, construire et stocker ce graphe est trop coûteux en place mémoire. Nous avons choisi une construction dynamique du graphe, mais malgré quelques tentatives nous n'avons pas trouvé d'heuristique plus performante que *best first*.

Le calcul d'erreur couleur produit un découpage avec des intervalles plus larges sur les extrémités du spectre, car les sensibilités couleurs décroissent rapidement. Cela explique la disparition des raies dans l'exemple 6. Par contre, le découpage dichotomique produit une représentation à 6 intervalles alors qu'un seul suffirait : la position de la raie influe sur le nombre de subdivisions nécessaires à son isolement.

4 Mise en œuvre dans un lancer de rayons

4.1 Construction des expressions

La méthode proposée a été ajoutée au logiciel de lancer de rayons *YART* [Roe93]. Chaque expression représente l'énergie collectée par l'arbre de rayons associé à un pixel. A chaque interaction d'un rayon avec un matériau, plusieurs opérations sont ajoutées à l'expression. Elles correspondent, par exemple, à la participation de l'énergie directe et de l'énergie réfléchie ou réfractée au point considéré. Lorsque la BRDF intervient dans le calcul d'une énergie, il faut évaluer le terme géométrique afin de construire sa représentation spectrale. On aurait pu calculer la représentation de tous les intervalles de la décomposition du domaine visible, mais il est plus efficace de créer un opérateur unaire qui calculera la représentation des intervalles nécessaires à l'évaluation adaptative à partir des coefficients C_i^j.

4.2 Erreur Couleur - Choix du seuil

Les erreurs couleurs sont exprimées dans l'espace XYZ. Pour imposer que la couleur calculée adaptativement soit perceptuellement indiscernable de la couleur de référence (obtenue par échantillonnage régulier), il faudrait utiliser un espace perceptuel comme CIELAB. La transformation de l'espace XYZ en CIELAB n'est pas linéaire et difficilement approximable. Nous avons préferé diviser le cube unitaire XYZ en cellules régulières, et estimer dans chaque cellule l'erreur XYZ correspondant à une erreur CIELAB fixée. L'algorithme d'intégration couleur devient :

```
évaluation pseudo-perceptuelle :
    calculer la racine de la représentation de chaque opération de l'expression
    calculer la couleur et l'erreur de la representation
    faire
        déterminer la cellule correspondant à la couleur
        récupérer le seuil associé à la cellule
        arrêter si l'erreur de la représentation est inférieure au seuil
        évaluer l'expression avec ce seuil
        (l'évaluation fournit la couleur et l'erreur associée)
    fin
```

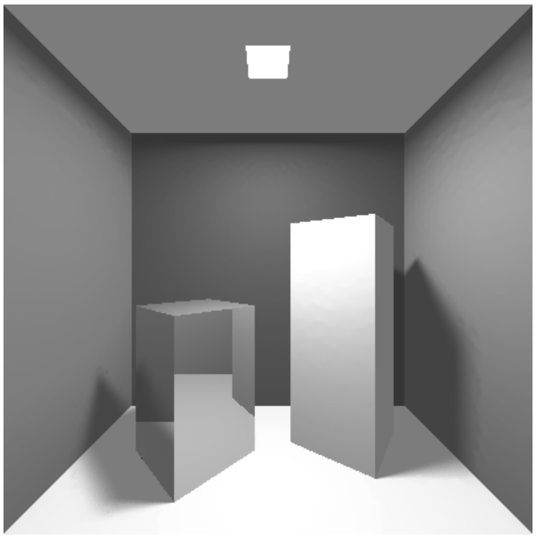

FIG. 7 – cornell box, *source surfacique D6500, seuil de 3*

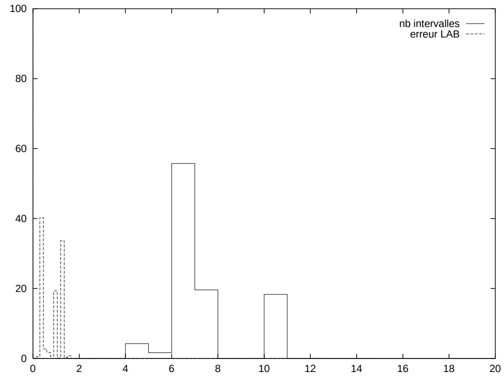

FIG. 8 – *distributions d'erreurs et d'intervalles pour un seuil de 3*

Des tests psychovisuels effectués dans le laboratoire [Alb98] ont montré qu'une erreur CIELAB inférieure à 5 unités est imperceptible dans les conditions normales d'utilisation d'un poste de travail.

5 Résultats

La figure 7 présente une classique *cornell box* avec une source surfacique de type D6500 et des murs dont les matériaux sont construits avec quelques réflectances tirées du ColorChecker [MMD76] et des BRDF simple isotropes [Sch94]. Les distributions des erreurs CIELAB et des nombres d'intervalles obtenues pour un calcul avec un seuil de 3 sont présentées à la figure 8.

Les différences entre deux images [Alb98] calculées avec des seuils de 3 et 10 sont présentées à la figure 9. Les différences sont calculées dans l'espace CIELAB et le niveau de gris correspond à l'erreur quadratique. Les zones noires correspondent à des erreurs inférieures à 3 et les zones blanches à des erreurs supérieures à 6. Les distributions des erreurs pour le seuil de 10 sont visibles à la figure 10. Les temps de calculs sont respectivement de 14mn46s, pour le seuil de 3, et de 9mn54s pour le seuil de 10. Le temps de calcul est bien fonction de la précision des calculs.

La même scène calculée en RGB prend 5mn32s et 9mn40s en spectral. Le rendu RGB de la scène comportant une source sodium n'est pas présenté puisque les erreurs ne sont pas visibles en noir et blanc. Cependant l'image présente de nombreuses aberrations comme des reflets verts sur un mur bleu éclairé par une source orangée.

La gestion des sources surfaciques utilise la méthode développée dans [ZSP98]. Les vecteurs lunineux directs [ZSP98] ont aussi été intégrés dans le rendu spectral, ce qui permet de gérer efficacement les sources surfaciques. L'utilisation des vecteurs lumineux oblige à stocker la représentation de l'énergie incidente. La décomposition complète du domaine visible doit être stockée pour le rendu spectral. Chaque spectre occuppe actuellement 6K octets. Le nombre de vecteurs lumineux à stocker varie entre 5000 et 10000 pour une scène aussi simple que la *cornell box*. Le rendu d'une scène plus complexe impose de stocker d'avantage de vecteurs lumineux. La représentation actuelle est donc trop coûteuse en place mémoire, mais de nombreuses économies sont possibles et devraient permettre de réduire l'encombrement de la décomposition complète à environ 2K octets.

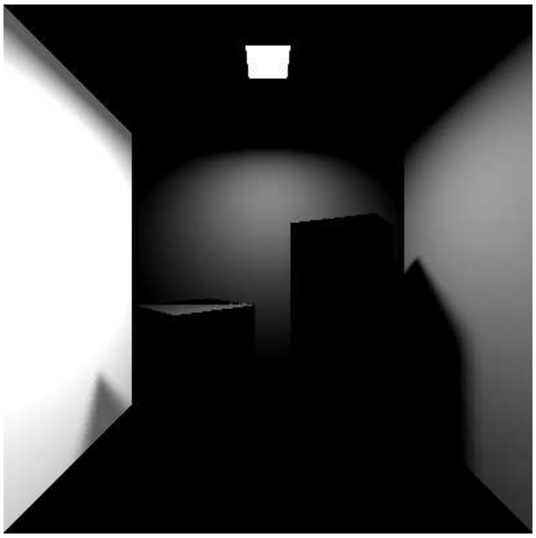

FIG. 9 – *image de distance (seuil 10 - seuil 3)*

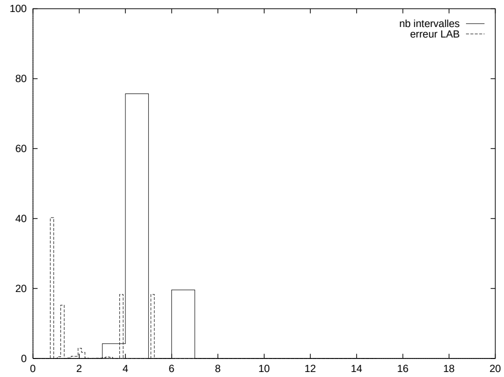

FIG. 10 – *distributions d'erreurs et d'intervalles pour un seuil de 10*

6 Conclusions - Perspectives

La représentation adaptative remplit bien les objectifs que nous nous étions fixés : l'estimation de l'erreur commise est suffisante et le temps de calcul augmente avec la précision requise. Cependant le calcul adaptatif est actuellement légèrement plus long que le calcul par échantillonnage régulier. La construction, le stockage et la maintenance de la structure dynamique associée aux expressions est une des premières causes du ralentissement. Il faut néanmoins tenir compte du fait qu'en réflexion, la géométrie est commune à toutes les longueurs d'ondes. Il reste maintenant à vérifier que l'exploitation de la cohérence de la réfraction d'un groupe d'ondes permettra de compenser ce ralentissement.

Les prochains développements vont concerner la réduction des temps de calculs et de l'occupation mémoire, l'intégration de l'éclairage indirect en utilisant les vecteurs lumineux de [ZSP98], le calcul des caustiques en réflexion, mais aussi des objets transparents et des caustiques dispersives. A plus long terme, l'intégration d'un modèle de vision permettrait de remplacer le contrôle actuel, local au pixel, par un autre plus global.

Références

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

[BM95] Mark R. Bolin and Gary W. Meyer. A frequency based ray tracer. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 409–418. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[BM98] Mark R. Bolin and Gary W. Meyer. A perceptually based adaptative sampling algorithm. In M. Cohen, editor, *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 299–309, July 1998.

[CLR92] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms.* MIT Press and McGraw-Hill Book Company, 6th edition, 1992.

[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.

[GH97] S. Gibson and R. J. Hubbold. Perceptually-driven radiosity. *Computer Graphics Forum*, 16(2):129–141, 1997. ISSN 0167-7055.

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

[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 G. Drettakis and N. Max, editors, *Eurographics Workshop on Rendering '98*, pages 223–236, June 1998.

[Nil82] Nils J. Nilsson. *Principles of Artificial Intelligence.* Symbolic Computation. Springer-Verlag, 1982.

[Pee93] Mark S. Peercy. Linear color representations for full spectral rendering. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 191–198, August 1993.

[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.

[Roe93] Marc Roelens. *Un Environnement pour le Tracé de Rayons utilisant une Modélisation par Arbre de Construction.* PhD thesis, Ecole des Mines de Saint-Etienne, France, February 1993.

[RP97] Gilles Rougeron and Bernard Péroche. An adaptive representation of spectral data for reflectance computations. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 127–138. Eurographics, June 1997. ISBN 3-211-83001-4.

[Sch94] C. Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum*, 13(3):C/233–C/246, 1994. EUROGRAPHICS'94.

[Ste94] A. Stentz. Optimal and efficient path planning for partially-known environments. In Edna Straub and Regina Spencer Sipple, editors, *Proceedings of the International Conference on Robotics and Automation. Volume 4*, pages 3310–3317. IEEE Computer Society Press, May 1994.

[Ste95] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1652–1659, August 1995.

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

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

[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.