Revue Électronique Francophone d'Informatique Graphique, Volume 8, Numéro 1, pp. 21–31, 2014

Rendu Progressif basé Metropolis-Hasting dans des scènes à contextes topologiques multiples

A. Gruson, [1] M. Ribardière, [2] R. Cozot [1] et K. Bouatouch[1]

[1]Université de Rennes 1, IRISA
[2]Université de Poitiers, XLIM-SIC

Résumé

Dans ce papier, nous proposons une nouvelle méthode de rendu basée sur Metropolis-Hasting et le lancer de photons stochastique. En s'appuyant sur la connaissance a priori de la topologie de la scène, notre méthode adapte automatiquement les paramètres de la marche aléatoire réalisée par l'algorithme de Metropolis-Hasting. Un taux d'échantillonnage optimal est ainsi obtenu pour chacun des contextes topologiques de la scène en leur associant une chaîne de Markov indépendante. Notre méthode peut être combinée avec toutes les avancées récentes concernant ce type d'algorithme (Replica Exchange, Adaptive Markov Chain Monte Carlo). En outre, notre papier propose une analyse détaillée des problèmes soulevés par l'utilisation de Metropolis-Hasting et montre comment notre méthode offre des résultats de qualité meilleure que celle obtenue avec les approches récentes du domaine.

Mots clé : Synthèse d'image, Rendu physiquement réaliste, Metropolis, Chaînes de Markov, Photon mapping, Rendu progressif

1. Introduction

La problématique de l'éclairage global s'intéresse au comportement de la lumière et de ses multiples réflexions dans une scène virtuelle, a fait l'objet de nombreuses études ces dernières années. Un grand nombre de techniques (tracer de chemins, lancer de photons, etc.) ont ainsi été développées. L'éclairage global se modélise mathématiquement sous la forme d'une intégrale. En conséquence, la plupart de ces techniques utilisent une approche de type Monte Carlo avec un échantillonnage par importance du domaine d'intégration.

Cependant, dans certains cas, l'échantillonnage par importance ne garantit pas un résultat satisfaisant pour des temps de calcul raisonnables. En effet, pour que cette stratégie soit efficace, les échantillons utilisés doivent être générés aléatoirement suivant une fonction de densité de probabilité (*fdp*) proportionnelle à la fonction à intégrer. Cette *fdp* n'étant pas connue a priori, il est difficile de choisir judicieusement celle-ci pour l'ensemble du problème à résoudre. Afin de palier ce problème, Veach et al. [Vea98] ont proposé une approche basée sur l'algorithme de Metropolis-Hasting. Cette dernière permet de construire à l'aide d'une marche aléatoire une densité de probabilité arbitraire.

Cette marche aléatoire ainsi que son paramétrage sont dépendants des contextes topologiques de la scène traversés par le chemin de lumière. Par *contexte topologique*, nous définissons des régions de la scène où les conditions d'éclairage (visibilité des sources de lumière) et le comportement lumineux en général (réflexions sur les surfaces notamment) sont relativement homogènes. Une scène peut être composée de plusieurs contextes topologiques comme, par exemple, plusieurs pièces dans un appartement avec une vue de l'extérieur à travers une fenêtre. Nous proposons une technique s'appuyant sur la connaissance a priori de ces contextes afin de mieux échantillonner le domaine du problème à résoudre et ainsi diminuer la variance de l'estimateur. Les principales contributions du papier sont :

- une analyse détaillée des problèmes soulevés par l'utilisation de l'algorithme de Metropolis-Hasting dans le cadre de simulation d'éclairage dans des scènes à contextes topologiques multiples.
- une utilisation de la connaissance de ces contextes pour mieux contrôler la marche aléatoire du Metropolis-Hasting.
- la détermination automatique du taux d'échantillonnage requis pour chaque contexte topologique (pièces intérieures, extérieurs, etc.)
- l'extension des améliorations récentes dans le domaine du Metropolis-Hasting appliquées au rendu (*Replica Exchange* et *Adaptive MCMC*) : cela permet de défi-

nir des paramètres propres à chaque contexte et donc d'optimiser l'échantillonnage.

La section 1 aborde les points théoriques concernant le Metropolis-Hasting pour la simulation d'éclairage. Les problèmes soulevés par une telle approche dans le cadre du rendu de scènes à contextes topologiques multiples y sont aussi analysés. Notre méthode est détaillée dans la section 3. Enfin, une étude comparative de notre approche (section 3.1) et les perspectives futures (section 4.2) sont ensuite présentées.

2. Analyse de la problématique

2.1. Le problème et sa formulation dans l'espace des chemins

La problématique de léclairage global a été formalisée par Kajiya et al. [Kaj86] avec ce qu'on appelle communément l'équation de rendu. Elle est exprimée sous la forme d'une intégrale sur le domaine des surfaces S de la scène :

$$L(x' \to x'') = L_e(x' \to x'') + \int_S L_i(x \to x') f_r(x \to x' \to x'') G(x \leftrightarrow x') dA(x) \quad (1)$$

où x, x', x'' sont des points sur l'ensemble des surfaces S, $L(x' \to x'')$ est la luminance arrivant du point x' vers x'', $L_e(x' \to x'')$ est la luminance émise au point x' vers x'', $L_i(x \to x')$ est la luminance incidente venant du point x vers x', $f_r(x \to x' \to x'')$ est la BSDF, $G(x \leftrightarrow x')$ est le terme géométrique et $dA(x)$ la mesure dans l'espace des surfaces. Cette équation a par la suite été exprimée dans l'espace des chemins par Veach [Vea98] :

$$I_j = \int_\Omega f_j(\bar{x}) d\mu(\bar{x}) \quad (2)$$

où I_j est la luminance au pixel j, Ω est l'espace des chemins, $\bar{x} = x_1 x_2 ... x_k$ est un chemin composé de $1 < k < \infty$ sommets (voir la figure 1) et l'intégrande $f_j(\bar{x})$ est égale à

$$f_j(\bar{x}) = L_e(x_1 \to x_2) G(x_1 \leftrightarrow x_2) W_e^{(j)}(x_{k-1} \to x_k) \prod_{i=2}^{k-1} f_s(x_{i-1} \to x_i \to x_{i+1}) G(x_i \leftrightarrow x_{i+1}) \quad (3)$$

où $W_e^{(j)}$ est l'importance émise par la caméra.

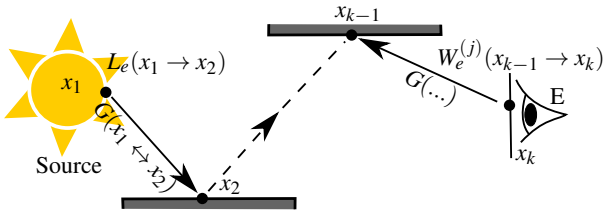


Figure 1: Formalisme utilisé pour décrire un chemin de lumière composé de k sommets : x_1 est un point sur la source lumineuse et x_k un point sur le plan image.

Pour calculer l'intégrale (eq. 2), on utilise une approche de Monte Carlo avec échantillonnage par importance :

$$< I_j >= \frac{1}{N} \sum_{i=0}^{N} \frac{f_j(\bar{x}_i)}{p(\bar{x}_i)} \quad (4)$$

avec $\bar{x}_i$ l'échantillon généré de façon aléatoire (un chemin de lumière) et $p(\bar{x}_i)$ la probabilité de générer cet échantillon. Différentes méthodes de rendu génèrent différents types de chemins (Path tracing, Light tracing, Photon mapping [Jen96], Bidirectionnal Path tracing [Vea98, LW93], ...) et la densité de probabilité est construite à l'aide d'une ou plusieurs stratégies d'échantillonnage. Ces différentes stratégies peuvent être combinées avec de l'échantillonnage par importance multiple (*MIS* pour "Multiple Importance Sampling" [Vea98]). Notons ici que la fonction de densité de probabilité utilisée pour l'échantillonnage par importance se base sur une connaissance partielle de l'intégrande (voir eq. 2). Par exemple, une première stratégie d'échantillonnage distribue les échantillons proportionnellement à la BSDF $f_r(x_2 \to x_1 \to x_0)$. Une autre stratégie permet de distribuer des échantillons sur les sources de lumière proportionnellement à la contribution de celles-ci. Dans ce cas, la densité ne prend en compte qu'une partie du facteur $L_i(x_2 \to x_1)$ (la composante directe). Pourtant, dans l'exemple d'une scène avec de l'éclairage complexe, la luminance incidente provenant de multiples réflexions peut être le facteur prédominant dans les variations de l'intégrande et donc créer un surplus de variance. Ainsi, le problème avec l'échantillonnage par importance est que la densité de probabilité n'est pas liée à l'ensemble de l'intégrande à évaluer.

2.2. L'algorithme de Metropolis-Hasting

L'algorithme de Metropolis-Hasting (Alg. 1) génère un échantillonnage dont la distribution est déterminée par une fonction d'importance $I(X_i)$. Plus la fonction d'importance aura une valeur élevée, plus l'échantillonnage sera dense. Par exemple, la probabilité d'être dans l'état X_i est égale à :

$$P^*(X_i) = \frac{I(X_i)}{b} \quad (5)$$

où $I(X_i)$ est la fonction d'importance scalaire et $b = \int_\Omega I(X) dX$, un facteur de normalisation (Alg. 1, ligne 1). P^* est la densité de probabilité dite *stationnaire* qui sera proportionnelle à $I(X)$ où X est un état représentant un chemin $\bar{x}$. Pour construire cette densité de probabilité, l'algorithme établit une marche aléatoire dans l'espace des chemins Ω. Cette marche est une chaîne de Markov qui utilise une loi de probabilité T (aussi appelée mutation) pour choisir, étant donné l'état actuel X_i à l'instant i, un nouvel état Y (appelé proposition) généré avec une probabilité $T(X_i \to Y)$ (Alg. 1, ligne 4) et qui est accepté avec une probabilité $a(X_i \to Y)$ comme nouvel état de la chaîne. Pour que l'algorithme converge vers la distribution stationnaire P^*, c'est-à-dire la solution à notre problème, l'algorithme de Metropolis-Hasting doit respecter les règles suivantes :

1. Règle de la *detailed balance* : $f(X)T(X \to Y)a(X \to Y) = f(Y)T(Y \to X)a(Y \to X)$. Celle-ci est respectée si on utilise $a = min(1, \frac{f(Y)T(Y \to X)}{f(X)T(X \to Y)})$ la probabilité de passer de l'état X à l'état Y. (Alg. 1, lignes 5 et 8..12)
2. L'ensemble des mutations doit permettre de couvrir l'ensemble du domaine où $f(X) > 0$, c'est-à-dire que $T(X \to Y) > 0$ pour toutes les paires d'état (X, Y) avec $f(X) > 0$ et $f(Y) > 0$.
3. Un *startup bias* peut survenir et doit être pris en compte. En effet, pour éviter ce problème il faut que la probabilité de sélectionner X_0, l'état initial de la chaîne de Markov, soit égale à $I(X)$. Dans sa thèse, Veach [Vea98] apporte une solution à ce problème (Alg. 1, ligne 2).

Ensuite, pour chaque étape de la marche aléatoire, il faut enregistrer la contribution de l'état courant dans le plan image. Pour cela, nous pouvons utiliser l'espérance mathématique pour enregistrer la contribution des échantillons générés (alg 1, lignes 6 et 7).

Il est à noter que l'utilisation de Metropolis-Hasting peut se traduire par une diminution de la vitesse de convergence de l'algorithme. En effet, dans le cadre de cet échantillonnage, les échantillons sont corrélés. Les échantillons indépendants d'un échantillonnage de Monte Carlo permettent de garantir que l'écart-type σ sera égal à $\sigma_p/\sqrt{N}$ avec N échantillons et σ_p l'écart-type de la variable aléatoire. Dans le cas de Metropolis, Kelemen et al. [KSKAC02] montrent que l'écart-type peut être borné suivant :

$$\sigma \leq \sigma_p \cdot \sqrt{\frac{1 + 2\sum_{k=1}^{N} R(k)}{N}} \qquad (6)$$

où $R(k)$ est la borne supérieure de la corrélation entre $I(X_i)$ et $I(X_{i+1})$. Ainsi, trop de corrélation dans la génération des chemins peut accroître la variance. Cela peut être soit la conséquence d'un taux d'acceptation trop élevé dû à de trop petites mutations, soit, la conséquence d'un taux d'acceptation trop petit entrainant de fait trop peu de changement d'état. On définit le taux d'acceptation comme la proportion d'états mutés ayant entrainé un changement d'état dans la chaîne de Markov.

Veach et al. [Vea98] utilisent comme espace de mutation l'espace des chemins. En ce sens, plusieurs types de mutation sont alors proposées pour garantir l'efficacité de la méthode (mutation bi-directionnelle, perturbation lentille, perturbation caustique, ...). D'autres auteurs ont également développé de nouveaux types de mutation pour des cas spécifiques [JM12].

Par la suite, Kelemen et al. [KSKAC02] ont proposé comme domaine de mutation l'hypercube des nombres aléatoires (figure 2), c'est-à-dire l'espace des nombres aléatoires permettant de générer un chemin donné. L'avantage d'utiliser cette formulation est double. Sa mise en œuvre est simple puisque cela consiste à transformer le générateur de nombre

Algorithm 1 Metropolis-Hasting

1: Évaluation du facteur de normalisation :$b \approx \int_\Omega I(X)dX$
2: $X_0 \leftarrow$Initialisation de l'état initial
3: **for** $i = 1...N$ **do**
4: $\quad Y \leftarrow$ Mutation(X_{i-1})
5: $\quad a \leftarrow min(1, \frac{I(Y)T(X_{i-1} \to Y)}{I(X_{i-1})T(Y \to X_{i-1})})$
6: $\quad$ EnregistrementContrib$(\frac{1-a}{N}\frac{F(X)}{I(X)/b})$
7: $\quad$ EnregistrementContrib$(\frac{a}{N}\frac{F(Y)}{I(Y)/b})$
8: $\quad$ **if** $a > rand(0,1)$ **then**
9: $\quad\quad X_i \leftarrow Y$
10: $\quad$ **else**
11: $\quad\quad X_i \leftarrow X_{i-1}$
12: $\quad$ **end if**
13: **end for**

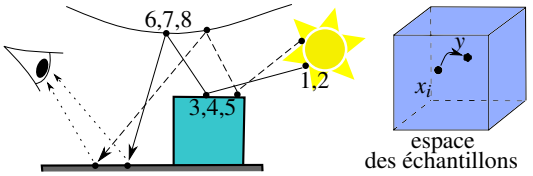


Figure 2: Dans Kelemen et al. [KSKAC02], l'espace des mutations est l'hypercube des nombres aléatoires (cube bleu). Les nombres aléatoires sont utilisés dans la création du chemin. Les chiffres $1, 2, .., 9$ sont les indices des nombres aléatoires générés. L'algorithme de Metropolis-Hasting va perturber ce vecteur de nombres aléatoires x_i en y puis utiliser ce vecteur pour tracer le nouveau chemin (en pointillé).

aléatoire pour qu'il puisse rejouer des séquences de nombres aléatoires. D'autre part, si l'échantillonnage par importance permet de réduire la dynamique de la fonction à intégrer, utiliser le domaine des nombres aléatoires permet de profiter de cette réduction. Seuls deux types de mutation sont alors nécessaires. Le premier utilise une perturbation dite "grande" pour garantir la couverture de l'ensemble du domaine d'intégration et réduire la corrélation entre les échantillons. Cette perturbation consiste alors à tirer uniformément de nouveaux nombres aléatoires. Le second utilise la perturbation suivante :

$$\Delta u = sign(\xi_1 - 0.5) r_2 \times exp(-ln(r_2/r1)\xi_2) \qquad (7)$$

avec $r_2 = 1/64$, $r_1 = 1/1024$, ξ_1, ξ_2 des nombres aléatoires générés uniformément entre $[0, 1]$ et Δu la variation de la valeur du nombre aléatoire. Hachisuka et al [HJ11] ont ensuite proposé une formulation différente pour qu'une perturbation avec un paramètre de plus grande valeur tende vers un échantillonnage uniforme dans le cadre de mutations très grandes.

Cline et al. [CTE05] ont proposé une autre formulation de Metropolis en redistribuant l'énergie à travers les pixels voisins avec les mutations de Veach. Par ailleurs, un travail récent [LKL*13] propose d'utiliser la mutation développée

par Jakob et al. [JM12] dans le gradient du domaine des chemins.

Pour résumer, Metropolis-Hasting permet d'échantillonner à l'aide d'une densité de probabilité P^* (eq. 5) définie par la fonction d'importance $I(X)$. Nous allons voir comment choisir judicieusement cette fonction d'importance dans la section suivante.

2.2.1. Fonction d'importance

Dans le cas de l'éclairage global, Veach [Vea98] et Kelemen [KSKAC02] utilisent une fonction d'importance $I(X)$ basée sur la luminance arrivant sur le plan image. Il y a plusieurs problèmes liés à l'utilisation de cette fonction d'importance. En effet, l'algorithme de Metropolis-Hasting distribue les échantillons proportionnellement à la distribution de la luminance, c'est-à-dire que les zones à faible luminance reçoivent très peu d'échantillons. Pour résoudre ce problème, Veach et al. [Vea98] proposent plusieurs solutions. Premièrement, pour éviter les fortes valeurs, les calculs se focalisent seulement sur l'éclairage indirect (la composante directe sera alors calculée avec une méthode de Monte Carlo classique). Deuxièmement, l'espérance de passer d'un état à un autre est utilisée pour générer des échantillons dans les zones à faible luminance (alg 1, lignes 6 et 7).

Les auteurs ont aussi proposé d'effectuer l'échantillonnage de Metropolis en deux étapes. Dans un premier temps, l'algorithme estime la luminance arrivant sur une image basse résolution. Dans la seconde étape, cette estimation est utilisée pour diviser la fonction d'importance, dans le but de la rendre plus plate. Le problème est alors que la première estimation peut être bruitée et donc produire une mauvaise fonction d'importance. Pour résoudre cela, Hoberock et al. [HH10] ont généralisé cette approche sous la forme d'un algorithme multi-passes en faisant croître de façon itérative la résolution du rendu. Ils utilisent aussi une fonction d'importance évaluée en fonction de la variance de la chaîne de Markov et d'une métrique perceptuelle.

Dans le cadre d'un *photon mapping* progressif, Hachisuka et al. [HJ11] utilisent la fonction de la visibilité d'un photon qui est égale à 1 si un photon du chemin est visible, sinon 0. L'utilisation d'une fonction d'importance binaire permet des simplifications dans les termes de Metropolis-Hasting. Chen et al. [CWY11] proposent quant à eux une fonction d'importance basée sur la densité de photon. Celle-ci est estimée lors d'un précalcul de quelques passes avant l'échantillonnage de Metropolis.

Il est assez commun que, dans la formulation des fonctions d'importance, certains types de chemins soient privilégiés, tels que les chemins de caustique.

2.2.2. Replica Exchange

Il est tout à fait possible d'utiliser plusieurs fonctions d'importance pour le même rendu. Chacune de ces fonctions d'importance est associée à une chaîne de Markov. Le *Replica Exchange* permet d'échanger l'état courant de différentes chaînes de Markov définies dans deux fonctions d'importance différentes. Pour que les conditions de l'algorithme de Metropolis restent valides, il faut que la probabilité d'échanger ces deux états soit égale à :

$$r(X_i, X_j) = \frac{I_i(X_j) I_j(X_i)}{I_i(X_i) I_j(X_j)} \tag{8}$$

où X_i et X_j sont des états de chaînes de Markov pour les fonctions d'importance I_i et I_j respectivement.

Kitaoka et al. [KKK09] proposent de se baser sur le *Replica Exchange* pour avoir plusieurs fonctions d'importance et permettre de contrôler le nombre d'échantillons pour chacune d'entre elles. Dans leurs travaux, les fonctions d'importance sont définies comme une partition des configurations possibles des chemins. Toutefois, cette technique est compliquée à mettre en œuvre pour le calcul du facteur de normalisation et la gestion du *startup bias*.

Dans un autre contexte, Hachisuka et al. [HJ11] utilisent le *Replica Exchange* comme la grande mutation telle que décrite dans [KSKAC02]. Par ailleurs, dû à l'utilisation d'une fonction d'importance égale à 1 quand le chemin est visible, le *Replica Exchange* sert aussi au calcul du facteur de normalisation b (Alg. 1, ligne 1).

Dans notre travail, nous allons utiliser plusieurs fonctions d'importance pour gérer nos différentes composantes spatiales. Pour chacune d'entre elles, nous aurons donc une chaîne de Markov que l'on va faire muter pour pouvoir effectuer une exploration locale de la fonction d'importance.

2.2.3. Adaptive Markov Chain Monte Carlo

Chaque chaîne de Markov doit effectuer des mutations pour explorer l'espace des états possibles. Cependant, la taille de mutation est un élément crucial pour l'efficacité de l'exploration. En effet, si une mutation est trop petite, cela va créer un taux de corrélation trop important et, inversement, si la mutation est trop grande, cela va créer trop de rejets dans la fonction d'acceptation. En mathématique, il existe une méthode appelée *Adaptive MCMC* [AT08] qui permet de contrôler la taille de la mutation par une autre chaîne de Markov :

$$\theta_{t+1} = \theta_t + H(t, \theta_t, X_t, ..., X_1) \tag{9}$$

où θ_{t+1} est la nouvelle taille de mutation et H la fonction de variation de la taille de la mutation. Pour que l'algorithme de Metropolis converge vers la densité P^*, il faut que :

$$lim_{t \to \infty} H(t, \theta_t, ...) = 0 \tag{10}$$

Hachisuka et al. [HJ11] utilisent cette méthode pour converger vers un taux d'acceptation optimal de $A^* = 23.4\%$ [RHB*08]. En effet, la taille de la mutation influence directement le taux d'acceptation. Le contrôle du taux d'acceptation permet de contrôler également la taille de la mutation.

Enfin, Yu-Chi et al. [LFCD07] utilisent une technique appelée *Population Monte Carlo* pour sélectionner une taille de mutation dans un ensemble de taille prédéfinie. Leur travail s'inscrit dans le contexte d'une redistribution de l'énergie à la manière de Cline et al. [CTE05].

2.3. Discussion

Les fonctions d'importance définies dans le plan image utilisent soit des informations de luminance soit se focalisent sur des chemins spécificiques (les caustiques par exemple). La fonction d'importance peut également évoluer au cours du rendu. Cela introduit malheureusement un surcout dû au calcul du facteur de normalisation (Alg. 1, ligne 1). Par ailleurs, une fonction d'importance constante permet une simplification de l'algorithme Metropolis-Hasting. Par exemple, Hachisuka et al. dans [HJ11] utilise la visibilité. Cependant, cette technique ne prend pas en compte le fait que certaines parties de la scène ont des topologies différentes. Ces topologies différentes créent un contexte d'échantillonnage particulier qui doit être pris en compte.

Par ailleurs, l'exploration locale des chaînes de Markov peut être controlée par la taille de la mutation ou par la fonction d'importance. Cependant, l'efficacité de cette exploration dépend du contexte d'échantillonnage. Plusieurs papiers se sont intéressés à l'extraction de la topologie spatiale de façon automatique [HDS03]. Dans le domaine du rendu, Fradin et al. [FMH05] ont utilisé une découpe spatiale pour réaliser un rendu "out-of-core" pour des scènes d'intérieur. De plus, les auteurs utilisent la notion de "portail" pour permettre le stockage et l'échange des flux entre les différentes topologies.

Nous proposons d'exploiter cette connaissance a priori de la topologie de la scène. Cette connaissance permet de définir les contextes d'échantillonnage et ainsi améliorer l'efficacité de l'exploration du domaine. Même si la détection automatique de ces contextes est une piste de travail intéressante, nous considérons actuellement qu'elle est connue a priori.

Notre méthode utilise une fonction d'importance dite *spatiale* adaptée à chaque contexte topologique. Combinée à des techniques de *Replica Exchange* et d'*Adaptive MCMC*, cette fonction d'importance permet de mieux définir le contrôle de l'échantillonnage des chemins de lumière dans la scène.

3. Fonction d'importance spatiale

Dans notre travail, nous avons décidé de nous baser sur les travaux d'Hachisuka et al. [HJ11] en s'inspirant de leur fonction d'importance exploitant la visibilité et en utilisant l'hypercube des nombres aléatoires pour éffectuer nos mutations [KSKAC02]. De plus, notre méthode est basée sur la méthode appelée "Stochastic Progressive Photon Mapping" (*SPPM*)" [HJ09] que nous décrivons rapidement ci-dessous. Pour plus d'informations, le lecteur est invité à lire l'article associé.

La méthode SPPM propose une version progressive de la méthode de lancer de photons [Jen96]. Des rayons primaires sont émis depuis la caméra pour chaque pixel de l'image. Lorsqu'un rayon primaire intersecte une surface de la scène, le point d'intersection est conservé dans une structure appelée *gather point*. Un disque d'influence, utilisé pour l'estimation de densité des photons, est associé à chacun des *gather points*. Ce disque permet de collecter l'ensemble des photons l'impactant. Ensuite, lors de la phase de lancer de photons depuis les sources de lumière, la luminance des *gather points* impactés par un photon est mise à jour. Il est alors possible de savoir, lors de la génération d'un chemin de lumière, si le chemin courant est contributif, c'est-à-dire qu'un *gather point* est proche d'un des sommets du chemin. Lorsqu'un lot de photons est complètement lancé, le disque d'influence des *gather points* est réduit. Cette réduction se base sur une analyse statistique de la densité de photons collectés pour chaque *gather point*. Le processus recommence alors en générant un nouveau lot de photons.

Lors de la phase de lancer des photons, les chemins de lumière traversent différentes conditions d'éclairage. L'idéal serait de prendre en compte ces variations de contexte topologique lors de l'échantillonnage. C'est ce que nous proposons avec notre méthode.

3.1. Notre méthode

Nous allons dans cette section décrire notre technique (voir l'Alg 2) et définir nos fonctions d'importance propres à chaque contexte topologique. Nous proposons une méthode généralisée à $N_{context}$ contextes. Nous appelons "chemin uniforme" un chemin généré par des nombres aléatoires indépendants de l'état de la chaîne de Markov.

Fonction d'importance Nous voulons prendre en compte les différentes difficultés des composantes spatiales par le biais de la définition de multiples fonctions d'importance. En effet, chacune d'entre elle ont des tailles de mutation différentes. Pour cela, nous modelisons chaque composante spatiale par une fonction d'importance qui lui est propre. Pour chaque composante spatiale j, nous proposons d'utiliser une nouvelle fonction d'importance dite spatiale $I_j(\bar{x})$ pour le chemin $\bar{x} = x_1 ... x_k$, définie comme :

$$I_j(\bar{x}) = \begin{cases} 1 & si\ \exists x_i\ tel\ que\ S_j(x_i) = 1 \\ 0 & sinon \end{cases} \quad (11)$$

où $S_j(x_i)$ est la visibilité spatiale du sommet x_i telle que

$$S_j(x_i) = V(x_i)E_j(x_i) \quad (12)$$

avec $V(x_i)$ la visibilité du sommet par rapport au *gather points* et $E_j(x_i) = 1$ si un des *gather points* recouvrant x_i se trouve dans la composante spatiale j sinon 0. Cette information de la composante spatiale est stockée directement

dans les *gather points* et est directement disponible pendant le tracer de photons. Cette information doit être redéfinie à chaque passe de génération des *gather points* (Alg.2, lignes 1-2 et 29-30). Dans la génération des *gather points*, on choisi de façon aléatoire la position dans le pixel pour génèrer un rayon et calculer l'intersection avec la scène. Par ailleurs, en partique, nous utilisons des volumes englobants pour définir les différentes composantes spatiales.

Startup bias (Alg.2, lignes 3-7) Par ailleurs, le problème du *startup bias* est facile à résoudre dans notre cas. En effet, puisque la fonction d'importance est constante, il suffit, pour initialiser l'algorithme, de trouver un chemin de lumière valide qui touche la composante spatiale j.

Startup bias Comme dans [HJ11], nous utilisons une fonction d'importance constante qui va nous permettre de générer des chemins uniformes. Pour chaque nouvel échantillon, un chemin uniforme est tracé (Alg.2, ligne 10) dans le but d'essayer de décorréler les échantillons entre eux. Ensuite, pour chaque composante spatiale j, la méthode détermine si cet échantillon uniforme est valide ou non, autrement dit si un de ses sommets est visible ou non (Alg.2, ligne 12). S'il est valide, alors il devient l'état courant de la chaîne de Markov pour la fonction d'importance j (Alg.2, ligne 14).

Mutation et Adaptive MCMC Si le chemin uniforme U_i n'est pas visible dans notre composante spatiale j, l'état courant de cette chaîne de Markov $X_{i,j}$, qui gère la composante spatiale j, est muté en utilisant une taille θ_j. Ce nouvel échantillon muté Y est accepté si il est visible dans le contexte j (Alg.2,lignes 15-24). Dans le même temps, la taille de mutation est mise à jour (Alg.2,ligne 23) :

$$\theta_j = \theta_j - \frac{(A^* - (MutCountAcc_i/MutCount_i))}{MutCount_i} \quad (13)$$

avec $A^* = 23.4\%$ le taux d'acceptation optimal, $MutCountAcc_i$ le nombre de chemins mutés acceptés et $MutCount_i$ le nombre total de chemins mutés.

Facteur de normalisation Il est calculé à l'aide des chemins uniformes

$$I_{j,c} = \int I_j(u)du \approx \frac{N_{I,I_j(u)=1}}{N_{I,total}} \quad (14)$$

où $N_{I,I_j(u)=1}$ est le nombre de chemins uniformes visibles dans la composante spatiale j et $N_{I,total}$ le nombre total des chemins uniformes tirés depuis les sources de lumière (Alg.2,lignes 13 et 16). $I_{j,c}$ est un facteur de normalisation qui permet, dans l'étape de collecte des informations sur les *gather points*, de mettre à l'échelle leur luminance (Alg.2,ligne 28).

Splatting et Rendu La contribution du chemin symbolisé par l'état courant de la chaîne de Markov $X_{i,j}$ ne prend en compte que les photons touchant la composante spatiale j (Alg.2,ligne 25). Lorsque l'étape de lancer des photons est finie, les statistiques des *gather points* (taille du disque d'influence, nombre de photons touchant le *gather point*, ...) sont mises à jour (Alg.2,ligne 28). Enfin, les derniers états de la chaîne de Markov sont sauvegardés dans l'état initial de celle-ci pour pouvoir continuer l'exploration locale de celle-ci lors de la prochaine passe de rendu (Alg.2,ligne 31).

Algorithm 2 Notre algorithme

```
 1: gps ← ReconstructionGP(gps)
 2: UpdateSpatialComp(gps)
 3: for j = 1...NContext do
 4:    X_{0,j} ←Initialisation état pour I_j
 5:    UniCount_j ← 1; UniCountAcc_j ← 1
 6:    MutCount_j ← 0; MutCountAcc_j ← 0; θ_j ← 1
 7: end for
 8: for t = 1...NPass do
 9:    for i = 1...NSamples do
10:       U_i ← TraceUniform()
11:       for j = 1...NContext do
12:          if I_j(U_i) == 1 then
13:             UniCount_j += 1; UniCountAcc_j += 1
14:             X_{i,j} ← U_i
15:          else
16:             UniCount_j += 1; MutCount_j += 1
17:             Y ← Mutate(X_{i-1,j}, θ_j)
18:             if I_j(Y) == 0 then
19:                X_{i,j} ← X_{i-1,j}
20:             else
21:                X_{i,j} ← Y; MutCountAcc_j += 1
22:             end if
23:             UpdateTheta(θ_j)
24:          end if
25:          Splat(X_{i,j}, j)
26:       end for
27:    end for
28:    UpdateGPAndDisplay()
29:    gps ← ReconstructionGP(gps)
30:    UpdateSpatialComp(gps)
31:    [X_{0,0}, ...] ← [X_{NSamples,0}, ...]
32: end for
```

4. Résultats et Discussion

Nous avons comparé notre méthode à celles de :

- *SPPM* : Méthode proposée par Hachisuka et al. [HJ09] qui utilise un échantillonnage uniforme (pas de Métropolis-Hasting).
- *VSPPM* : Méthode proposée par Hachisuka et al. [HJ11] qui utilise Metropolis-Hasting et une fonction d'importance basée sur la visibilité.
- *ISPPM* : Méthode proposée par Chen et al. [CWY11] qui utilise Metropolis-Hasting et une fonction d'importance basée sur l'exponentielle inverse de la densité de

photon. Ici, la fonction d'importance est plus forte dans les zones ayant reçu peu de photons.

Toutes les scènes de test ont été rendues sur un Intel(R) Xeon(R) CPU E5640 @ 2.67GHz en utilisant tous les coeurs disponibles. A noter que nous avons implémenté l'ensemble des techniques dans le moteur de rendu *Mitsuba* [Jak10].

Par ailleurs, pour plus d'impartialité, nous avons intégré un *Adaptive MCMC* et le *Replica Exchange* [HJ11] dans la technique de Chen et al. [CWY11] car nous savons que le taux d'acceptation, le nombre de chemins uniformes et l'expression de la perturbation sont des éléments importants dans la vitesse du taux de convergence des algorithmes de Metropolis-Hasting. Cependant, nous ne nous intéressons pas à cet aspect et cela fera l'objet de travaux futurs. Pour la configuration des algorithmes, nous avons utilisé uniquement les paramètres par défaut fournis par les auteurs. Le calcul des disques d'influence des *gather points* est réalisé de la même manière pour toutes les techniques.

Par ailleurs, nous utilisons deux métriques dans cet article. En plus d'une RMSE, nous utilisons une RMSE pseudo-perceptuelle $RMSE_{log}$

$$RMSE_{log}(P,P_{ref}) = \sqrt{\frac{\sum_{i=1}^{N_p}(log(p_i)-log(p_{ref,i}))^2}{N_p}} \quad (15)$$

avec p_i et $p_{ref,i}$ les valeurs RGB du pixels j dans les images P et P_{ref} respectivement et N_p étant le nombre total de pixels. En effet, le système visuel humain est plus sensible à l'intensité relative qu'à sa valeure absolue. Les experimentations connues sous le nom de "Threadhold versus Intensity" (TVI) [EGP*10] ont montré que la perception de la luminance par le système visuel humain est quasi linéaire dans le domaine logarithmique.

Dans les sections suivantes, nous discuterons des résultats obtenus avec un cas d'étude simplifié. Ensuite, nous analyserons des résultats obtenus dans le cas d'une scène complexe.

4.1. Analyse d'un cas d'étude simplifié

Dans un premier temps, nous étudions le cas d'une simple scène avec deux contextes distincts éclairés par une carte d'environnement (voir figure 3). Pour chacun des contextes, nous pouvons faire l'analyse suivante :

1. Le contexte extérieur : l'éclairage direct est facile à calculer et l'algorithme de Metropolis-Hasting n'est pas obligatoirement nécessaire. Des techniques utilisant un échantillonnage uniforme ou des algorithmes de Metropolis-Hasting avec de grandes mutations vont explorer efficacement l'espace des chemins visibles.
2. Le contexte intérieur : l'éclairage direct est difficile à calculer puisque les chemins de lumière doivent passer par les ouvertures (fenêtres). La plupart des chemins contributifs ont alors subi plusieurs réflexions (éclairage indirect). De plus, il est peu probable de trouver un chemin uniforme contributif pour ce contexte. Dans ce cas, Metropolis-Hasting est une solution intéressante. Cependant, il est à noter que la luminance moyenne dans ce contexte est plus faible que dans le contexte d'extérieur.

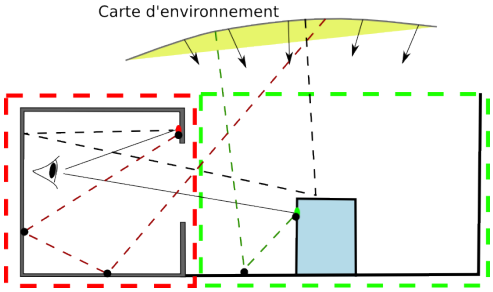


Figure 3: Notre cas d'étude avec deux contextes : le contexte intérieur encadré par le carré rouge et le contexte extérieur encadré par le carré vert. La scène est éclairée par une carte d'environnement. Les chemins de lumière (en pointillé) peuvent impacter un seul des contextes ou bien les deux.

Les résultats sont affichés dans la figure 4. Les tailles de mutation, le taux d'acceptation et la RMSE pour les deux contextes et pour chacune des techniques sont récapitulés dans le tableau 1.

Méthode	Taille mutation	τ_{acc}	RMSE
VSPPM			
Contexte int.	0.19034	5.91%	1.43×10^{-3}
Contexte ext.	0.19034	23.52%	5.3×10^{-3}
ISPPM			
Contexte int.	0.156191	7.73%	0.89×10^{-3}
Contexte ext.	0.156191	26.91%	6.1×10^{-3}
Multi.			
Contexte int.	0.073768	23.47%	0.45×10^{-3}
Contexte ext.	0.191576	23.24%	8.2×10^{-3}

Table 1: Statistiques pour les différents algorithmes avec comme informations par contexte : la taille de mutation, le taux d'acceptation (τ_{acc}) et la RMSE pour le cas d'étude (figures 3 et 4).

VSPPM On remarque ici que pour le contexte extérieur, c'est la méthode qui offre la plus faible RMSE. En effet, comme la fonction d'importance est la même et que le contexte extérieur est plus facile à échantillonner, ce dernier reçoit une grande partie des échantillons. C'est pour cela aussi que la taille de la mutation est plus grande que pour les autres techniques. Par ailleurs, un problème dans la formalisation de Hachisuka et al. est que le *Replica Exchange* provoque un suréchantillonnage des zones les plus faciles. En effet, après avoir exploré un contexte difficile à échantillonner, le *Replica Exchange* avec le chemin uniforme va avoir une forte probabilité de trouver un chemin contributif

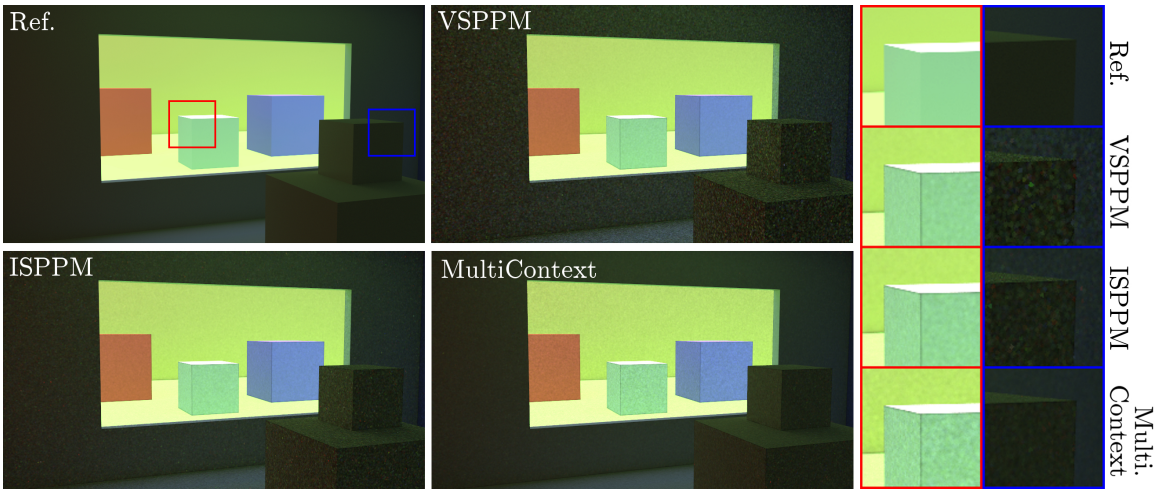


Figure 4: *A Gauche :* comparaison d'images produites par les différentes techniques après 10 minutes de calcul. *A Droite :* zooms sur chacun des contextes.

dans une zone facile, ce qui va arrêter l'exploration locale dans les zones difficiles.

ISPPM Cette technique permet par rapport à VSPPM de mieux placer les échantillons dans le contexte intérieur (RMSE plus faible) mais cela a un impact sur la RMSE du contexte extérieur. Par ailleurs, la taille de la mutation est plus petite que celle de VSPPM puisque l'exploration locale des chaînes de Markov a plutôt lieu dans le contexte intérieur.

Notre méthode La taille de mutation est différente pour chacun des contextes. Elle est plus petite pour le contexte intérieur que pour les deux autres techniques. Ceci est dû à la difficulté d'échantillonnage. Par ailleurs, il faut noter :

- une amélioration significative du bruit et de la RMSE dans le contexte intérieur. En effet, la moitié des échantillons sont utilisés dans ce contexte. De plus, la taille de mutation apprise par *Adaptive MCMC* permet d'avoir un taux d'acceptation dans ce contexte proche du taux d'acceptation optimal.
- une dégradation de la RMSE et une augmentation du bruit dans le contexte extérieur. En effet, les échantillons sont répartis équitablement. Cependant, il est à noter que la taille de la mutation est un peu plus grande que la taille apprise par les autres techniques.

Notre technique permet un apprentissage local de la taille de la mutation. Celle-ci permet d'avoir un taux d'acceptation pour les différentes chaines de Markov proche du taux d'acceptation optimal. Par ailleurs, notre méthode permet un taux d'échantillonnage équilibré entre les deux contextes.

4.2. Scène complexe

La scène "Breakfast" est composée d'un :

- contexte extérieur : c'est la salle du fond qui est éclairée par une carte d'environnement. La lumière passe par une large baie vitrée. Cependant, seulement 0.02 % de chemins uniformes arrivent dans cette zone.
- contexte intérieur : c'est la salle où se trouve la caméra. Celle-ci est éclairée de manière indirecte par la salle du fond, la lumière passant par une large ouverture. Elle est aussi éclairée par des petites fenêtres obstruées par des persiennes. Dans ce contexte, seulement 0.0014 % des chemins uniformes sont visibles.

Les résultats pour la scène "Breakfast" sont présentés dans la figure 5. Par ailleurs, les différentes courbes de RMSE sont disponibles dans la figure 6 et la $RMSE_{log}$ dans la figure 7. Il est facile de remarquer que, dans le contexte intérieur, notre méthode produit des résultats beaucoup moins bruités. Cependant, cette efficacité se paie par un peu plus de bruit dans le contexte extérieur. Ce bruit est visuellement mineur par rapport au gain obtenu dans le contexte intérieur comme le démontrent les courbes des figures 6 et 7.

5. Conclusion et perspectives

Nous avons montré dans notre analyse les problèmes soulevés par l'utilisation des méthodes de rendu basées sur l'algorithme de Metropolis-Hasting, notamment celui concernant la fonction d'importance. Celle-ci est dépendante du contexte dans lequel l'algorithme est exécuté. Malheureusement, une scène soumise à des conditions d'éclairage complexes est souvent composée de plusieurs contextes. Notre méthode associe donc, pour chacun de ces contextes, une

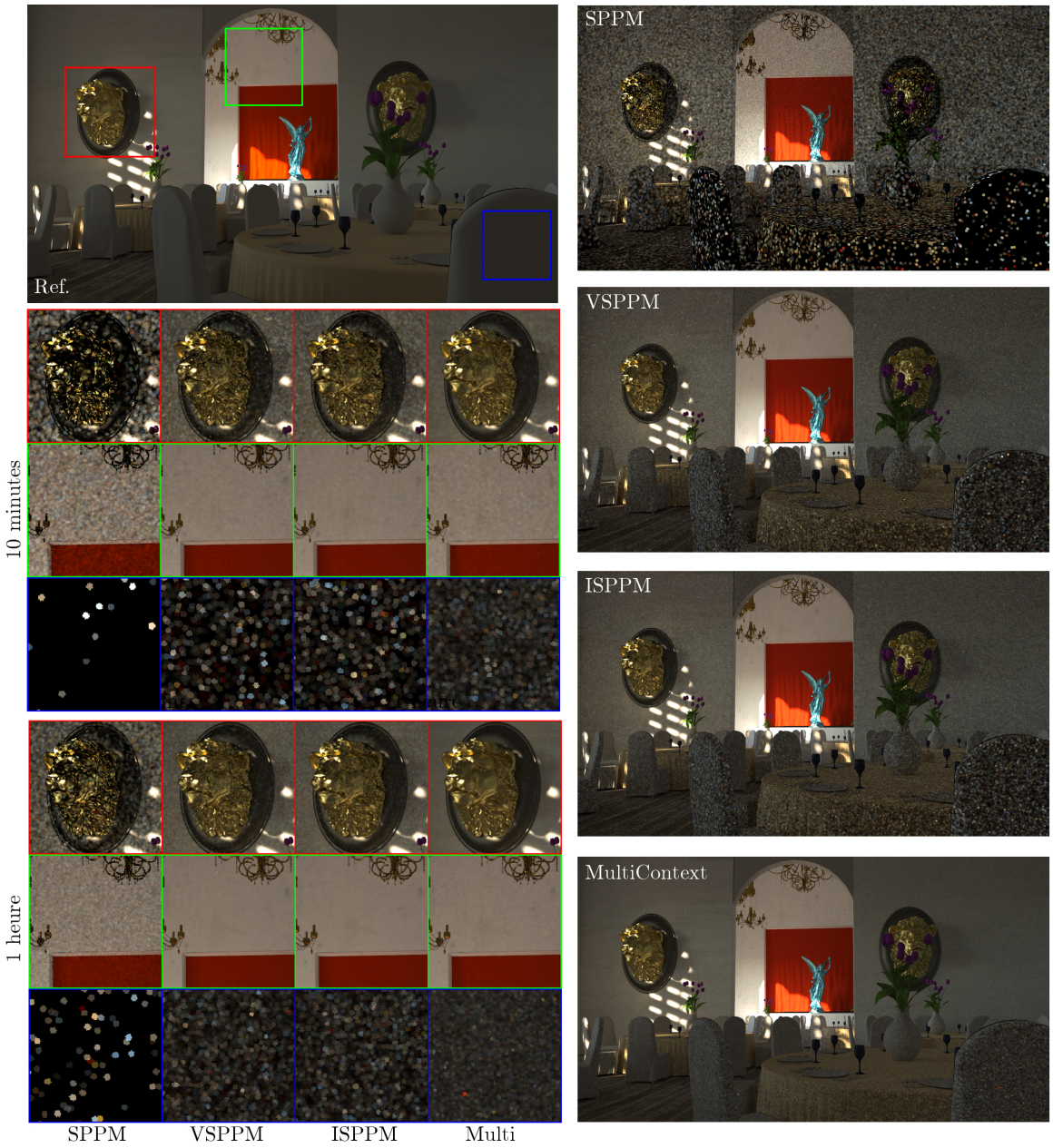
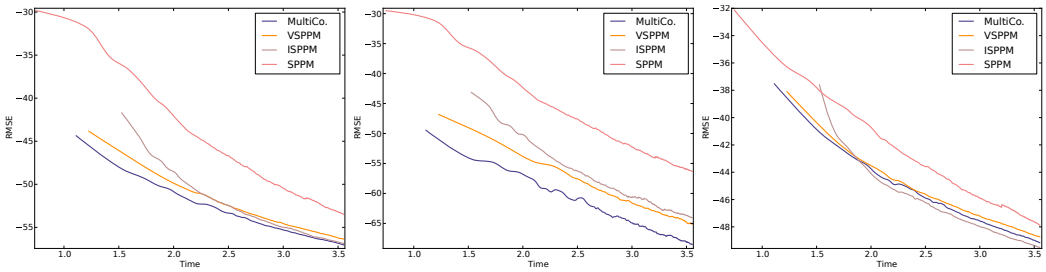


Figure 5: Résultats pour la scène "Breakfast" (par Greg Zaal). *Image du haut :* Image de référence calculée par un tracé de chemins avec 64k échantillons. *Colonne de droite :* Résultats avec les différentes méthodes après 1 heure de rendu. Ces différents rendus ont la même exposition (x 2^6)

fonction d'importance différente. Elle peut être combinée avec les avancées récentes du domaine telles que le *Replica Exchange* et l'*Adaptive Markov Chain Monte Carlo*. La qualité des résultats fournis valide parfaitement notre hypothèse et surpasse ceux obtenus avec les méthodes récentes. Ces résultats nous encouragent à poursuivre nos travaux dans ce sens. Nous souhaitons ainsi généraliser notre approche pour d'autres types d'algorithme de rendu (bidirectional path tracing, par exemple) et exploiter l'idée de contexte topologique pour le rendu de milieux participatifs. Pour le moment, la définition des contextes topologiques se fait à la main lorsque l'artiste modélise la scène. Cette étape peut être



(a) RMSE sur toute l'image (b) RMSE sur le contexte intérieur (c) RMSE sur le contexte extérieur

Figure 6: L'évolution de la RMSE pour la scène "Breakfast" au cours du temps, en échelle logarithmique. On remarque que notre méthode permet un gain significatif dans le contexte intérieur mais est légerement désaventagée dans le contexte intérieur. De plus, on observe, dans le contexte intérieur, que la technique ISPPM produit des résultats moins bons que VSPPM. Cela est dû au fait que, dans cette scène, la fonction d'importance de ISPPM est bruitée et produit un mauvais échantillonnage du contexte intérieur.

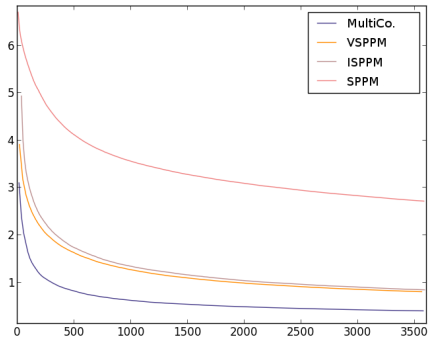


Figure 7: L'évolution de la $RMSE_{log}$ au cours du temps. Notre technique (courbe bleu) permet d'avoir une erreur largement inférieure aux autres techniques pour la scène fig. 5

fastidieuse et nécessite une bonne connaissance du comportement des algorithmes de rendu. Nous souhaitons pouvoir automatiser cette partie de notre algorithme en exploitant notamment la disposition des *gather points*.

Remerciements

Nous voulons remercier Ronan Boitard pour son aide précieuse et Jaroslav Křivánek pour ses remarques pertinentes. Par ailleurs, nous voulons remercier aussi les reviewers de REFIG qui ont permis d'améliorer ce papier.

Références

[AT08] ANDRIEU C., THOMS J. : A tutorial on adaptive mcmc. *Statistics and Computing. Vol. 18*, Num. 4 (2008), 343–373.

[CTE05] CLINE D., TALBOT J., EGBERT P. : Energy redistribution path tracing. *ACM Trans. Graph.. Vol. 24*, Num. 3 (juillet 2005), 1186–1195.

[CWY11] CHEN J., WANG B., YONG J.-H. : Improved stochastic progressive photon mapping with metropolis sampling. *Computer Graphics Forum. Vol. 30*, Num. 4 (2011), 1205–1213.

[EGP*10] ERIK R., GREG W., PAUL D., SUMANTA P., WOLFGANG H., KARO L. M. : *High Dynamic Range Imaging, 2nd edition*. Morgan Kaufmann Publishers, 2010.

[FMH05] FRADIN D., MENEVEAUX D., HORNA S. : Out-of-core photon-mapping for large buldings. In *Proceedings of Eurographics symposium on Rendering* (June 2005).

[HDS03] HAUMONT D., DEBEIR O., SILLION F. : Volumetric cell-and-portal generation. *Computer Graphics Forum. Vol. 22*, Num. 3 (2003), 303–312.

[HH10] HOBEROCK J., HART J. C. : Arbitrary importance functions for metropolis light transport. *Computer Graphics Forum. Vol. 29*, Num. 6 (2010), 1993–2003.

[HJ09] HACHISUKA T., JENSEN H. W. : Stochastic progressive photon mapping. *ACM Trans. Graph.. Vol. 28*, Num. 5 (décembre 2009), 141 :1–141 :8.

[HJ11] HACHISUKA T., JENSEN H. W. : Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph.. Vol. 30*, Num. 5 (octobre 2011), 114 :1–114 :11.

[Jak10] JAKOB W. : Mitsuba renderer, 2010. http ://www.mitsuba-renderer.org.

[Jen96] JENSEN H. W. : Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96* (1996), Springer-Verlag, pp. 21–30.

[JM12] JAKOB W., MARSCHNER S. : Manifold explora-

tion : a markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.. Vol. 31*, Num. 4 (juillet 2012), 58 :1–58 :13.

[Kaj86] KAJIYA J. T. : The rendering equation. *SIGGRAPH Comput. Graph.. Vol. 20*, Num. 4 (août 1986), 143–150.

[KKK09] KITAOKA S., KITAMURA Y., KISHINO F. : Replica exchange light transport. *Computer Graphics Forum. Vol. 28*, Num. 8 (2009), 2330–2342.

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F. : A simple and robust mutation strategy for the metropolis light transport algorithm. *Computer Graphics Forum. Vol. 21*, Num. 3 (2002), 531–540.

[LFCD07] LAI Y.-C., FAN S. H., CHENNEY S., DYER C. : Photorealistic image rendering with population monte carlo energy redistribution. In *Proceedings of the 18th Eurographics conference on Rendering Techniques* (Aire-la-Ville, Switzerland, Switzerland, 2007), EGSR'07, Eurographics Association, pp. 287–295.

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T. : Gradient-domain metropolis light transport. *ACM Trans. Graph.. Vol. 32*, Num. 4 (2013).

[LW93] LAFORTUNE E. P., WILLEMS Y. D. : Bi-directional path tracing. In *COMPUGRAPHICS 93* (1993), pp. 145–153.

[RHB*08] ROSENTHAL J. S., H C. F. M., BROOKS S., GELMAN A., JONES G., L. MENG X. : Optimal proposal distributions and adaptive mcmc, 2008.

[Vea98] VEACH E. : *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162.