



Comment simuler efficacement les interactions lumière/matière en synthèse d'image

Adrien Gruson et Kadi Bouatouch¹

La synthèse d'image a connu une évolution fantastique ces deux dernières décennies et son champ d'application s'est fortement élargi. Toutes les méthodes et tous les algorithmes dérivés de la synthèse d'image sont maintenant utilisés dans des domaines autres que celui de l'image tels que l'infrarouge, le son, la téléphonie, etc. Le réalisme des images de synthèse ne cesse de croître grâce aux nombreuses méthodes sophistiquées de rendu capables de simuler les diverses interactions lumière-matière comme : la réflexion, la réfraction, les diffusions sous-surfaciques ou dans un milieu semi-transparent (appelé aussi milieu participant). Ces différentes interactions sont prises en compte dans divers chemins de lumière qui connectent les sources de lumière à la caméra (ou observateur). La majorité des méthodes de rendu, utilisées dans le domaine du jeu vidéo et du cinéma, sont basées sur une approche stochastique comme la méthode de Monte Carlo. Cet article met l'accent sur cette classe de méthodes permettant la création d'image toujours plus proche de la réalité.

Principes et définitions

La synthèse d'image a connu une évolution significative durant ces dernières décennies. En effet, il y a peu de temps c'était un ensemble de techniques permettant de produire différents effets (perspective, ombres, transparence, textures...) plus ou moins fidèlement. Elle s'appuyait uniquement sur des modèles empiriques dont le seul but était d'obtenir une représentation de ces effets sans chercher à les reproduire fidèlement. Le but recherché était alors de produire des images *réalistes*.

1. IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France. E-mail : adrien.gruson@irisa.fr, kadi.bouatouch@irisa.fr.

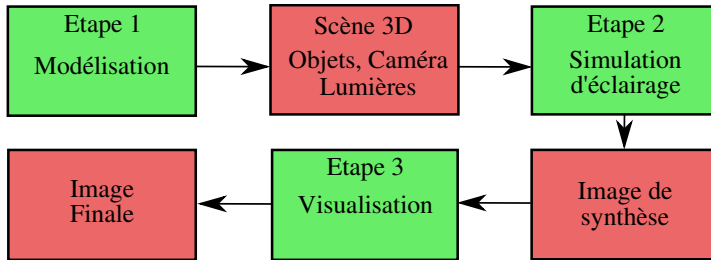


FIGURE 1. Différentes étapes (boîtes vertes) de la synthèse d'image. Chaque étape produit une donnée (image ou scène, boîtes rouges) qui est ensuite utilisée dans l'étape suivante.

Maintenant, la synthèse d'image a pris une direction plus scientifique en s'appuyant sur différents domaines de la science tels que la perception visuelle, la physique et les mathématiques. L'objectif actuel est de reproduire le plus précisément possible la réalité. Il s'agit donc de produire des images *photo-réalistes* où un utilisateur ne pourrait pas distinguer ce que produit un ordinateur de la réalité. Le processus de création d'une image de synthèse comprend trois parties (Figure 1) : la modélisation géométrique, la simulation des interactions lumière-matière (ou technique de rendu ou simulation d'éclairage) et la visualisation.

La modélisation géométrique consiste d'une part à créer des objets et d'autre part à les placer dans l'environnement 3D. Pour chaque objet, il faut définir ses paramètres géométriques (liste de sommets pour un polygone, normale...) et ses paramètres d'aspect (réflectivité, transparence, textures, couleurs...). Par ailleurs, l'utilisateur place dans la scène des sources de lumière et une caméra virtuelle. Cette caméra virtuelle, qui représente l'observateur, est associée à un écran virtuel (grille de petites surfaces carrées appelées pixels à travers lesquels sont déterminés les objets visibles). Elle a pour but de fixer une direction de vue formée par l'observateur et un point de visée (Figure 2).

Après avoir finement défini cet environnement (ou scène) 3D, la simulation de l'éclairage est réalisée pour produire l'image de synthèse. Le but de cette étape est de calculer la répartition de l'énergie lumineuse reçue par l'observateur à travers chaque pixel de l'écran virtuel. Cette énergie lumineuse est une grandeur photométrique appelée luminance et qui est associée à une distribution spectrale $L(\lambda)$, où λ est la longueur d'onde. L'image de synthèse obtenue est alors représentée par un tableau à deux dimensions de luminances. Pendant cette étape, l'ordinateur calcule toutes les interactions entre les sources de lumière et les objets de la scène. Ces interactions sont multiples (réflexion, réfraction, diffusion) et difficiles à évaluer pour l'ordinateur. C'est pour cette raison qu'il faut parfois plusieurs minutes voire des heures pour générer une image de bonne qualité.

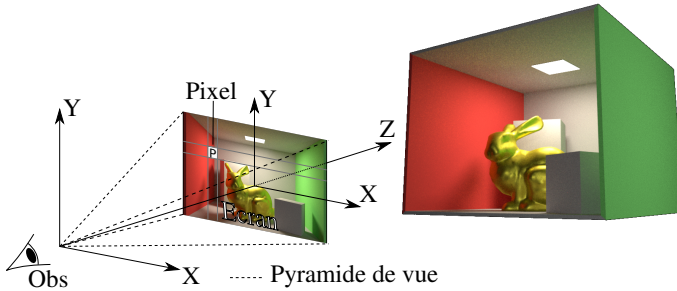


FIGURE 2. Repère de l'observateur et pyramide de vue (caméra)

Enfin, il nous faut passer par l'étape de visualisation. Cette étape va consister à convertir l'image physique en RVB (rouge, vert, bleu). Le but étant de pouvoir afficher l'image de synthèse sur un écran classique d'ordinateur.

Dans cet article nous ne parlerons que de la génération d'images de synthèse et pas de séquence vidéo. Cependant, pour générer une vidéo, il faut décrire le mouvement des objets de la scène, de la caméra, des sources de lumière et les changements d'aspect de ces objets dans le temps. Tout objet mobile est caractérisé par une trajectoire spatio-temporelle et des déformations plus ou moins complexes. Ensuite, cette scène animée est envoyée au logiciel de simulation d'éclairage qui va décomposer cette animation en une séquence d'images de synthèse (50 par seconde).

Dans la suite de cet article, nous présentons les différents types d'interaction lumière-matière qui sont pris en compte dans l'équation de luminance (aussi appelée l'équation de rendu). Puis, nous présentons les méthodes stochastiques de rendu permettant de résoudre l'équation de luminance afin de produire une image de synthèse photo-réaliste. Enfin, nous présentons l'étape de visualisation de ces images.

Simulation d'éclairage ou technique de rendu

Une source de lumière est considérée comme un rayonnement électromagnétique défini par un spectre, représentant une énergie lumineuse qui est une fonction de la longueur d'onde. La lumière émise par une source lumineuse interagit avec les objets d'un environnement, elle est soit réfléchiée, soit réfractée, soit diffusée par un milieu participant (fumée, poussière, feu, brouillard, nuage).

Pour atteindre le photo-réalisme, les efforts de recherche ont porté principalement sur l'élaboration de modèles d'éclairage permettant de rendre ces effets. L'équation de rendu [3] modélise ces effets :

$$L_r(\vec{x}, \omega_r) = \int_{\Omega} f_r(\vec{x}, \omega_r, \omega_i) \cdot L_i(\vec{x}, \omega_i) \cdot |\omega_i \cdot N| \cdot d\omega_i$$

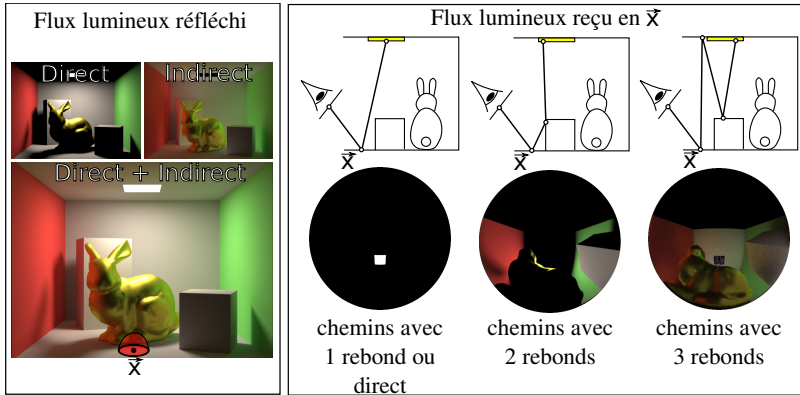


FIGURE 3. Les différents types de chemins lumineux. À gauche : les images dues à l'éclairage direct, indirect et la combinaison des deux. À droite : un aperçu de la luminance incidente au point \vec{x} pour différentes longueurs de chemin (nombre de rebonds).

où L_r est la luminance réfléchie au point \vec{x} vers l'observateur représenté par la direction ω_r . Cette luminance réfléchie est égale à l'intégrale sur l'ensemble Ω des directions incidentes ω_i au point \vec{x} . Pour chaque direction d'incidence ω_i , il faut connaître la luminance incidente L_i et le matériau au point \vec{x} . En effet, lors d'une interaction lumière-matière, seulement une certaine quantité de cette luminance incidente est réfléchie/réfractée vers l'observateur et dépend du matériau (décrit par le terme $f_r(\vec{x}, \omega_r, \omega_i)$). $|\omega_i \cdot N|$ est le cosinus de l'angle d'incidence entre ω_i et la normale N au point \vec{x} .

En ce qui concerne la luminance incidente en \vec{x} , elle est la conséquence de plusieurs facteurs qui sont résumés dans la figure 3. Premièrement, les sources de lumière peuvent éclairer directement \vec{x} . Si on ne prend en compte que cette contribution lumineuse et que l'on calcule le flux réfléchi en tout point visible de la scène, cela nous donne une image due à l'éclairage *direct* (Figure 3). Deuxièmement, il faut prendre en compte les contributions des autres objets de la scène (aussi appelées éclairage indirect). Ces contributions correspondent au fait que tous les objets sont éclairés par les sources de lumière et réfléchissent vers \vec{x} . On parle de chemin de lumière indirect généré après plusieurs rebonds de lumière (Figure 3 avec 2 et 3 rebonds). Pour générer une image de synthèse, il faut combiner ces deux contributions (indirect et direct).

En ce qui concerne les matériaux, ils permettent de modéliser l'aspect de l'objet. Par exemple, le fait que l'objet soit rouge en dessous d'un éclairage blanc est dû à son matériau. Par ailleurs, un matériau définit la proportion de lumière réfléchie vers

l'observateur. Il existe plusieurs modèles modélisant cette proportion, en voici les principaux :

— **Réflexion diffuse**

Une lumière incidente arrivant sur un objet mat pénètre dans celui-ci puis ressort avec la même amplitude quelle que soit la direction d'observation (de réflexion), c'est la réflexion diffuse. C'est par exemple le cas des murs de couleur dans la scène de la figure 3.

— **Réflexion spéculaire (objets brillants)**

Une lumière incidente arrivant sur un objet lisse (réfléchissant) est réfléchi par la surface de celui-ci mais uniquement dans certaines directions autour de la direction de réflexion idéale. Cette dernière est telle que l'angle de réflexion est égale à l'angle d'incidence. Par exemple, un miroir est quasiment parfaitement spéculaire, il n'y a qu'une seule direction de réflexion spéculaire qui est celle de la réflexion idéale. Dans la scène de la figure 3, le lapin est constitué d'un matériau brillant.

En résumé, le matériau, associé à un objet, caractérise la capacité de l'objet à réfléchir, réfracter et diffuser la lumière.

Méthodes de rendu stochastiques

Nous avons vu précédemment que les effets lumineux sont modélisés par l'équation de rendu. Cependant, nous avons vu que la luminance incidente est la contribution de chemins lumineux de différentes longueurs (ou nombre de rebonds). Une façon de considérer le problème de la simulation d'éclairage est de trouver l'ensemble des chemins lumineux qui contribuent à la luminance des pixels de l'écran de la caméra virtuelle. Plus formellement, on peut définir un chemin lumineux comme $\bar{x} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n)$ où \bar{x}_i est le $i^{\text{ème}}$ sommet (point d'intersection entre un rayon lumineux et un objet de la scène) composant ce chemin. La luminance I_j d'un pixel j (de l'écran) est égale à :

$$I_j = \int_P f_j(\bar{x}) d\mu(\bar{x}),$$

où P est l'ensemble des chemins lumineux et f_j est la fonction de contribution du chemin \bar{x} au pixel j . Cette fonction prend en compte les différentes valeurs des matériaux des objets avec lesquels le chemin lumineux interagit (rebond) et l'énergie émise par la source de lumière. Cette fonction peut être égale à zéro si le chemin n'apporte pas d'énergie lumineuse au pixel p .

En informatique, on sait évaluer la valeur d'une intégrale de grande dimension. Cependant, plus l'espace d'intégration est grand, plus son évaluation est coûteuse. Le problème est, qu'en synthèse d'image, nous avons affaire à des intégrales de très grande dimension. Monte Carlo est une technique qui permet d'approcher la valeur de l'intégrale (l'ensemble des chemins) en se basant uniquement sur un échantillonnage de l'espace des chemins. L'idée est de ne pas considérer tous les chemins de

lumière possibles mais uniquement prendre un échantillon de chemins possibles. Si l'échantillon est assez grand (beaucoup de chemins) et représentatif, alors on peut l'utiliser en considérant qu'il représente la totalité des chemins lumineux. Dans une approche Monte Carlo, on parle d'estimateur et celui-ci a la forme suivante :

$$E[I_j] = \frac{1}{N} \sum_{i=1}^N f_j(\bar{x}_i) / p(\bar{x}_i),$$

où N est le nombre de chemins utilisés dans l'échantillon et \bar{x}_i un chemin lumineux construit de manière aléatoire. La densité de probabilité $p(\bar{x}_i)$ permet de construire les chemins lumineux. En effet, l'idée est de construire des chemins qui contribuent à l'image finale et d'éviter les chemins non contributifs. La précision de l'estimateur (son erreur) va dépendre du nombre d'échantillons (N) utilisés et de la qualité des chemins générés ($p(\bar{x}_i)$). Augmenter le nombre d'échantillons fait croître de façon linéaire le temps de calcul. Donc, la seule façon d'avoir un calcul plus efficace est de recourir à de bonnes méthodes de construction des chemins lumineux.

Pour résumer, le but de la simulation d'éclairage est de construire des chemins lumineux contributifs. Ces chemins sont construits par morceaux en lançant des rayons. Un rayon est défini par sa position dans l'espace 3D et sa direction. Un des modules basique du moteur de rendu est celui qui calcule l'intersection de ce rayon avec les objets 3D de la scène. Ci-après, nous présentons différentes façons de construire des chemins lumineux.

Chemins générés à partir de la caméra

Une première possibilité est de générer un chemin lumineux à partir de la caméra (Figure 4). Pour commencer, on lance un rayon à partir d'un observateur à travers un pixel de l'écran virtuel vers les objets de la scène. À chaque intersection (point \bar{x} dans la figure 3), on veut calculer la contribution des sources de lumière. Pour ce faire, on va choisir de façon aléatoire un point sur une des sources de lumière et on lance un rayon dans sa direction. Si le rayon arrive à destination, alors on sait que le point est visible et on ajoute sa contribution. Cependant, cette approche permet de construire uniquement des chemins *directs*. Pour des chemins plus longs (indirects de longueur N comme dans le figure 3), nous devons étendre le chemin actuel. Pour ce faire, nous lançons un nouveau rayon depuis le dernier point d'intersection et dans une direction aléatoire. Ce nouveau rayon génère un nouveau point d'intersection et on recommence ce processus à partir de ce nouveau point (sélection d'un point sur une source de lumière, direction de rebond choisie de façon aléatoire). Pour éviter de construire des chemins de longueur infinie, avant de faire rebondir un rayon, un test aléatoire permet de décider de l'arrêt ou non de ce processus.

Chemins générés à partir d'une source de lumière

Une seconde possibilité est de générer un chemin lumineux à partir d'une source lumineuse (Figure 4). On commence par échantillonner, de manière aléatoire, un

point sur une source de lumière. Puis, un rayon est tracé dans une direction aléatoire. En chaque intersection, on essaye de déterminer si ce point est visible par l'observateur. Pour ce faire, on génère un rayon partant de notre point d'intersection actuel vers l'observateur. Si ce rayon touche l'écran de la caméra, alors on ajoute la contribution de ce chemin lumineux au pixel impacté.

Discussion

Nous avons vu deux façons de construire des chemins lumineux. En synthèse d'image, il existe une multitude d'algorithmes pour construire ces chemins. L'un des plus avancés est le tracé de chemin bidirectionnel où l'on trace deux chemins lumineux en même temps : un chemin depuis la caméra $\bar{x} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n)$ et un depuis la source de lumière $\bar{y} = (\bar{y}_0, \bar{y}_1, \dots, \bar{y}_n)$. Ensuite on essaye de connecter les sommets de ces deux chemins deux à deux. Pour chaque couple de sommets, on génère un rayon pour tester si les deux sommets sont mutuellement visibles ou non. S'ils sont visibles alors nous avons construit un chemin lumineux contributif. La figure 4 résume les différentes approches discutées et les images générées par ces différents algorithmes pour un temps donné (pour comparer l'efficacité de l'un par rapport à l'autre). Par ailleurs, dans cette figure, les images de synthèse sont générées avec différentes techniques de rendu. Ces images nous permettent de juger des différentes techniques : moins il y aura d'artefacts visuels (bruit, objets noirs), plus la méthode de rendu sera jugée efficace.

Metropolis-Hasting

Dans les approches vues précédemment, quelle que soit la technique de rendu choisie, la construction des chemins repose sur des décisions aléatoires (sélection d'un point sur une source de lumière, d'une direction de rebond). Dans la pratique, pour chaque chemin lumineux, un ensemble de nombres aléatoires est généré puis utilisé par la méthode de rendu pour construire un chemin lumineux. Cet ensemble contient des nombres aléatoires indépendants et, à chaque nouveau chemin lumineux, un nouvel ensemble est créé. Cela signifie que la génération d'un chemin n'est pas basée sur les chemins précédemment créés.

Pendant, dans des scènes complexes (lampes extérieures et observateur à l'intérieur d'un bâtiment, de multiples rebonds de lumière nécessaires), il est difficile de générer un chemin contributif. Donc, si par chance, nous réussissons à générer un chemin contributif (appelé aussi *chemin courant*), alors nous voulons l'utiliser pour explorer *localement* l'espace des chemins. Tout ceci dans l'espoir de trouver de nouveaux chemins contributifs proches de ce chemin. L'algorithme de Metropolis-Hasting [2, 6] nous permet cela en générant une marche aléatoire dans l'espace des chemins. À chaque étape de cette marche, il génère un chemin *proposé* \bar{y} en appliquant une mutation (petit changement) aléatoire au chemin \bar{x}_{i-1} qui est notre chemin courant. Si ce nouveau chemin \bar{y} contribue à l'image, il est alors accepté (après calcul

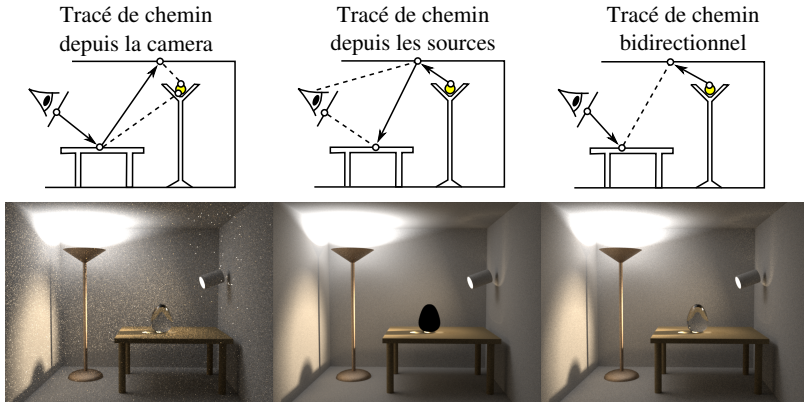


FIGURE 4. Cette image résume les différentes techniques décrites dans cet article. Chaque technique construit différemment les chemins lumineux et génère une image de synthèse (même temps de génération pour ces trois images). Ces différences impactent l'efficacité des algorithmes pour gérer tel ou tel phénomène lumineux. Ce manque est visible dans les images finales sous forme de bruit ou d'effets ignorés (objet noir dans la technique du centre).

de probabilité d'acceptation) et devient le nouveau chemin courant. Sinon, le chemin courant \bar{x}_i est égal à \bar{x}_{i-1} (donc pas de changement de chemin courant) et une autre mutation aléatoire est appliquée à \bar{x}_i .

À la fin de cet algorithme, nous obtenons une séquence de chemins $\bar{x}_0, \bar{x}_1, \dots, \bar{x}_N$ où chaque chemin dépend du chemin précédent. Ceci diffère des approches classiques basées sur Monte Carlo où les chemins sont indépendants entre eux. Enfin, pour générer l'image de synthèse, on utilise cette séquence de chemins dans un estimateur de Monte Carlo.

La scène représentée par la figure 5 permet de voir l'amélioration qu'apportent les algorithmes de rendu basés sur Metropolis-Hasting. Dans cette scène, il est difficile de trouver les chemins de lumière passant par la fenêtre. L'algorithme de Metropolis permet d'explorer localement cette fenêtre avec son mécanisme de mutation. Ceci impacte fortement l'efficacité de l'algorithme de rendu et produit des images sans bruit, ce qui n'est pas le cas des techniques de rendu n'utilisant pas Metropolis (appelées dans cette figure *uniforme*). La figure 6 montre d'autres exemples de rendu basés sur Metropolis-Hasting.



FIGURE 5. Scène Vitraux : éclairage à travers les fenêtres subissant une absorption et une diffusion par la poussière. Pour cette scène, l’algorithme de Metropolis a été utilisé. En effet, il est difficile de trouver les chemins lumineux passant par la fenêtre. Par ailleurs, nous montrons des zooms pour comparer la technique avec et sans (appelée uniforme) Metropolis. On observe une nette amélioration due à l’exploration locale faite par Metropolis.

Visualisation

Comme il a été mentionné précédemment, le processus de synthèse d’image fournit, pour chaque pixel, la luminance d’un point de la scène vue par l’observateur à travers ce pixel. Cette luminance dépend de la longueur d’onde (notons que dans le cas d’une approche trichromatique, trois longueurs d’onde sont utilisées : Rouge, Vert, Bleu). Appelons cette luminance $L(\lambda)$, λ étant la longueur d’onde. Rappelons que le spectre visible correspond à l’intervalle des longueurs d’onde [380 nm, 780 nm]. Pour pouvoir visualiser ces luminances sur un écran vidéo il va falloir les convertir en trois composantes Rouge, Vert, Bleu (RVB). Cette conversion est donnée par :

$$R = \int_{380 \text{ nm}}^{780 \text{ nm}} L(\lambda) \bar{r}(\lambda) d\lambda$$

$$V = \int_{380 \text{ nm}}^{780 \text{ nm}} L(\lambda) \bar{v}(\lambda) d\lambda$$

$$B = \int_{380 \text{ nm}}^{780 \text{ nm}} L(\lambda) \bar{b}(\lambda) d\lambda$$

Les fonctions $\bar{r}(\lambda)$, $\bar{v}(\lambda)$, $\bar{b}(\lambda)$ sont appelées fonctions de mélange trichromatiques. Elles dépendent du moniteur utilisé.

Pour pouvoir à la fois calculer précisément les luminances et les couleurs RVB associées, il est important de considérer un nombre de longueurs d’onde suffisant et

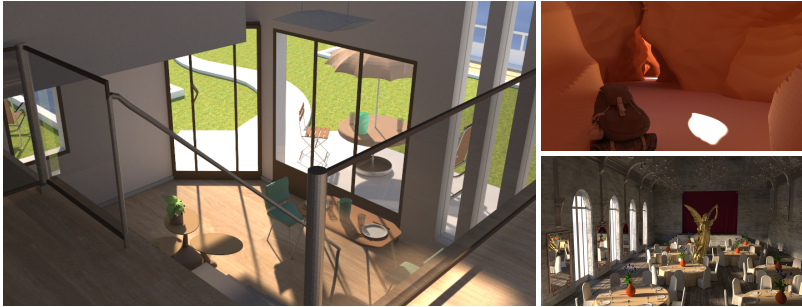


FIGURE 6. D'autres exemples d'images de synthèse générées avec un algorithme basé sur Metropolis.

des valeurs de longueur d'onde adéquates tel qu'il est montré dans les travaux de Meyer [4]. Notons qu'en synthèse d'image on utilise en général uniquement trois longueurs d'onde (Rouge, Vert, Bleu), ce qui évite cette conversion.

Pour les moniteurs classiques, chaque composante R, V, B est codée sur 8 bits (valeur comprise entre 0 et 256). Cependant, en synthèse d'image, nous n'avons pas cette limitation et effectuons la simulation avec une meilleure précision. D'autre part, nous n'avons pas de restriction sur la dynamique (rapport entre la valeur la plus basse par rapport à la valeur la plus haute) de l'image. Cependant, la luminance maximale affichée sur un moniteur est de l'ordre de 100 à 300 candelas par mètre carré alors que la luminance du ciel peut être de l'ordre de quelques milliers de candelas par mètre carré. Ce qui pose des problèmes de restitution de l'image calculée. La solution la plus simple, adoptée couramment, est de tronquer les valeurs ou d'effectuer une mise à l'échelle. Cette solution donne des résultats très insuffisants étant donné la perte d'information induite. Il est donc important de comprimer les valeurs trichromatiques RVB calculées en prenant en compte les propriétés du système visuel humain et du moniteur utilisé. Cette compression est appelée *Tone Mapping* [1, 5].

Conclusion

La synthèse d'image est passée d'un stade empirique à celui des outils et méthodes rigoureux utilisés dans diverses branches de la science. En effet, elle s'appuie sur l'informatique, la physique, la perception visuelle, la photométrie, la colorimétrie et les mathématiques. Autant de domaines qui ne peuvent que témoigner de l'intérêt et de la richesse scientifique de ce merveilleux domaine de recherche qu'est la synthèse d'image.

L'objectif majeur est de reproduire fidèlement et efficacement l'interaction entre la lumière, émise par des sources lumineuses, et les objets composant une scène 3D. Fidèlement veut dire bien simuler les lois de la physique et surtout prendre en compte

les caractéristiques du système visuel humain. Reproduire efficacement consiste à calculer la luminance d'un pixel de manière précise et avec un coût de calcul raisonnable. Ces deux exigences sont maintenant possibles grâce aux algorithmes basés sur les méthodes de Monte Carlo et de Metropolis-Hasting.

Références

- [1] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced high dynamic range imaging : theory and practice*. CRC Press, 2011.
- [2] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1) :97–109, 1970.
- [3] James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.
- [4] Gary W Meyer. Wavelength selection for synthetic image generation. *Computer Vision, Graphics, and Image Processing*, 41(1) :57–79, 1988.
- [5] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging : acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.
- [6] Eric Veach. *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford University, 1997.