



# DIPLÔME NATIONAL DE DOCTORAT

(Arrêté du 25 mai 2016)

Date de la soutenance : **7 mars 2018**

Nom de famille et prénom de l'auteur : **PERRIER Hélène**

Titre de la thèse : « Échantillonneurs Anti-aliassés pour du rendu photo-réaliste à base d'estimateurs de Monte-Carlo ».



## Résumé

### 1. Problématique

Lorsque l'on affiche un objet 3D sur un écran d'ordinateur, on transforme cet objet en une image 2D, c.a.d en un ensemble de pixels colorés. Les couleurs observées par l'œil humain sont fonction du déplacement de la lumière dans le monde et de la façon dont cette lumière interagit avec les différents matériaux des différents objets. On appelle *Rendu* la discipline qui consiste à trouver la couleur à associer aux pixels pour représenter une scène donnée.

Pour ce faire, on va placer une caméra virtuelle dans une scène 3D, découper le film de cette caméra en pixel, et calculer la quantité de lumière qui arrive sur chaque pixel du film (nommé aussi plan image) de cette caméra.

Si on nomme  $I$  le pixel du plan image qui nous intéresse, on calcule sa couleur en résolvant l'équation suivante (équation simplifiée)

$$I = \int_S L(p, v) V_p(v) dp$$

Ce que cette équation signifie c'est que pour trouver  $I$ , on va intégrer toute la lumière que chaque point  $p$  de la scène  $S$  renvoie dans la direction  $v$ ,  $v$  étant la direction menant de  $p$  à  $I$ . Le tout est pondéré par une fonction binaire  $V_p$  déterminant si un point  $p$  de  $S$  est visible ou non depuis  $I$ . Cependant pour déterminer la quantité de lumière que  $p$  renvoie vers  $v$ , on utilise la fonction  $L$ , qui elle-même intègre la totalité de la lumière que  $p$  reçoit depuis tous les autres points de la scène et calcule quelle fraction de cette lumière  $p$  renvoie vers  $v$ .

On a donc déjà deux intégrales dans cette version très simplifiée de l'équation. Malheureusement, l'ordinateur ne sait pas calculer des intégrales on a donc deux solutions possibles pour faire un rendu :

- Trouver une expression analytique qui permet de supprimer l'intégrale de l'équation (approche basée statistique).
- Approximer numériquement l'équation en tirant des échantillons aléatoires dans le domaine d'intégration et en en déduisant la valeur de l'intégrale via des méthodes dites de Monte Carlo.

La difficulté d'une représentation statistique de la surface réside dans le fait qu'une représentation statistique globale n'est pertinente que dans des cas très restreints (distribution gaussiennes, pas de problèmes de visibilité, uniformité de l'objet ...).

Lors de cette thèse nous nous sommes donc plutôt intéressés à la deuxième solution, celle de l'intégration numérique, basée sur la théorie de l'échantillonnage. Ce sont ces méthodes qui constituent le noyau dur de cette thèse, et qui ont données lieu au développement de deux nouveaux échantillonneurs. Notre première méthode génère des échantillons 2-D ayant une excellente distribution spatiale. Elle est étendue dans notre deuxième méthode, qui peut produire des échantillons N-D ayant une distribution de qualité équivalente. Cette seconde méthode permet également de contrôler la quantité d'échantillons localement et en offre la possibilité de rajouter incrémentalement des échantillons tout en préservant la qualité de la distribution.

Dans la suite, nous présenterons tout d'abord la théorie de l'échantillonnage, avant de détailler nos contributions.

## 2. Théorie de l'échantillonnage

Dans le cadre du Rendu, si on ne peut pas calculer analytiquement l'intégrale, on va approximer sa valeur simulant un certain nombre  $n$  de rayons lumineux. Cela revient donc à un problème d'échantillonnage; le problème étant de choisir correctement ces rayons pour avoir la meilleure approximation possible de l'intégrale.

Pour ce faire, on va utiliser des méthodes dites de Monte Carlo et résoudre l'équation :

$$E(I) = \sum_{i=1}^n p_i f(x_i)$$

Dans cette équation,  $x_i$  est le  $i$ ème échantillon, et  $f$  est la fonction à échantillonner. Par rapport à notre première  $f(x_i)$  correspondrai à  $L(x_i, v)V_p(v)$ . On a également  $\sum_{i=1}^n p_i = 1$  pour normaliser le tout. Plus  $n$  est grand, plus on est proche de la valeur réelle pour  $I$ .

On note que suivant le choix des échantillons  $x_i$  le résultat de cette équation varie. Dans le cas du rendu, le choix d'un échantillonneur est essentiellement fonction de cette variance. En effet, comme on intègre chaque pixel individuellement, plus cette variance est grande, plus on va observer de différences entre deux pixels voisins. Cela crée des artefacts dits de *bruit* dans l'image finale.

Il a été montré que le fait d'avoir une distribution spatiale des  $x_i$  de type *bruit bleu* est optimale pour réduire cette variance. Cette distribution, essaye de maintenir la même distance entre chaque point et tous ses voisins. Le plus souvent, on caractérise une distribution spatiale de points par son spectre de Fourier, qui est sa représentation fréquentielle.

Une autre solution pour réduire ce bruit est d'utiliser des échantillonneurs dits *basse discrèpance*.

La discrèpance est une mesure mathématique qui permet d'évaluer l'uniformité d'une distribution spatiale de points. Plus la discrèpance est basse, plus la distribution est uniforme. Pour être qualifié de *basse discrèpance*, un échantillonneur doit voir sa discrèpance baisser avec l'augmentation du nombre  $N$  de points à une vitesse  $O(\log(N)^{D-1}/N)$  où  $D$  est le nombre de dimensions.

Ces échantillonneurs présentent des distributions de points très structurées, et ont pour but d'avoir la distribution la plus uniforme possible. Malheureusement, lors d'un rendu, ils créent une sorte de grille très régulière dans l'image. C'est ce que l'on appelle *l'aliasage*. Ils ont cependant l'avantage pour la plupart d'être des séquences de points. Cela signifie que l'on peut rajouter des points à un ensemble existant tout en préservant la basse discrèpance.

Les travaux qui aujourd'hui constituent le noyau dur de ma thèse, consistent à concilier ces deux catégories d'échantillonneurs pour avoir une la variance la plus faible possible en intégration sans créer d'aliasage dans l'image finale.

Nous avons pu produire deux travaux sur ce domaine. Le premier présente un premier échantillonneur en 2 dimensions qui ne présente pas d'aliasage tout en ayant une faible variance en intégration. Il est décrit Section 3. Section 4 présente notre deuxième contribution.

Nous y présentons un échantillonneur toujours à faible variance et anti-aliasé, mais pour de plus hautes dimensions. Il permet également de contrôler localement la quantité d'échantillons et offre la possibilité d'augmenter incrementalement le nombre d'échantillons.

### 3. Premier échantillonneur 2D

Ce premier échantillonneur était la première tentative pour concilier à la fois une discrédance basse et une distribution *bruit bleu*. En utilisant des ensembles de points 1-D connus pour être basse discrédance, on peut construire un ensemble 2-D où chaque ligne et chaque colonne est elle-même un ensemble basse discrédance.

Il a été prouvé qu'un tel ensemble 2-D est lui-même un ensemble basse discrédance. Ensuite, en réarrangeant correctement cet ensemble 2-D, on peut changer la distribution des points. Cela nous permet d'obtenir une distribution de points bruit bleu (ou d'autres distributions), sans changer la discrédance de l'ensemble initial.

Pour réarranger l'ensemble de points on permute les coordonnées  $x$  des points puis les coordonnées  $y$ . Intervertir des points formant un ensemble basse discrédance 1-D ne change pas la discrédance de l'ensemble, du coup notre ensemble 2-D résultant est toujours composé de lignes et de colonnes qui sont-elles même basses discrédance.

Ce qui fait que notre ensemble final est basse discrédance, tout en ayant une distribution de points différente.

Pour obtenir une distribution ciblée, on précalcule et on stocke l'ensemble de permutations sur  $x$  et  $y$  à appliquer et l'échantillonneur n'a plus qu'à lire ces données et à faire la permutation. Tout ceci est illustré Figure 1.

Cette méthode a le mérite d'avoir réussi à prouver que la basse discrédance et la distribution bruit bleu étaient parfaitement compatibles. Elle est cependant limitée dans ses applications pratiques. Elle ne génère que des points 2-D, n'est pas séquence, et ne permet pas de contrôler localement la densité des points.

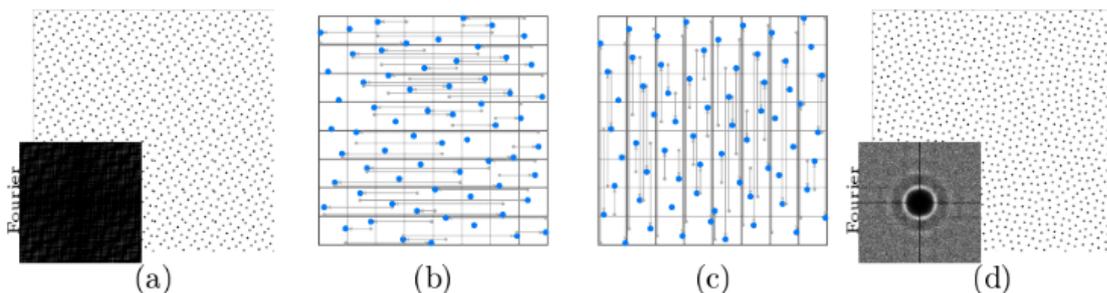


FIGURE 1 – (a) Présente l'ensemble de points 2-D initial de notre méthode. Cet ensemble est permuté sur chaque coordonnées (b,c) Pour donner un ensemble qui soit bruit bleu à l'arrivée (d). On peut noter que l'ensemble de départ a un spectre de Fourier très différent de celui de l'ensemble d'arrivée. On peut également noter que le spectre de Fourier de l'ensemble d'arrivée présente une croix en son centre. Cette croix est symptomatique de la basse discrédance et n'est en aucun cas un handicap aux capacités de notre échantillonneur en terme d'intégration.

### 4. Second échantillonneur nD et séquentiel

Ce second échantillonneur est lui aussi *bruit bleu* et basse discrédance. Il exploite le fait que si un ensemble est *bruit bleu* dans chaque sous partie du domaine, il le sera également sur le domaine entier.

Notre méthode part donc d'une séquence basse discrédance qu'elle subdivise en tuiles carrées. Ensuite, on applique nos permutations locales, sur chaque tuile. Nous avons prouvé formellement dans le papier que ces permutations nous permettent de préserver la discrédance de l'ensemble entier malgré le fait qu'elles soient locales. Parmi toutes les permutations locales possibles, nous avons précalculé celles qui correspondaient au mieux à une distribution *bruit bleu*, et nous les avons stockées.

De plus, grâce à des permutations globales, dont nous avons également prouvé qu'elles préservent la discrédance, nous pouvons rendre notre échantillonneur séquentiel. Cela signifie si nous avons un ensemble de points répartis dans des tuiles à un niveau  $l$ , nous pouvons resubdiviser ces tuiles au niveau  $l + 1$  pour rajouter des points tout en préservant la position des points déjà présents.

Enfin, comme notre système est basé sur des tuiles, on peut choisir de ne subdiviser qu'une partie des tuiles pour changer localement la densité des points. Tout ceci est illustré Figure 2.

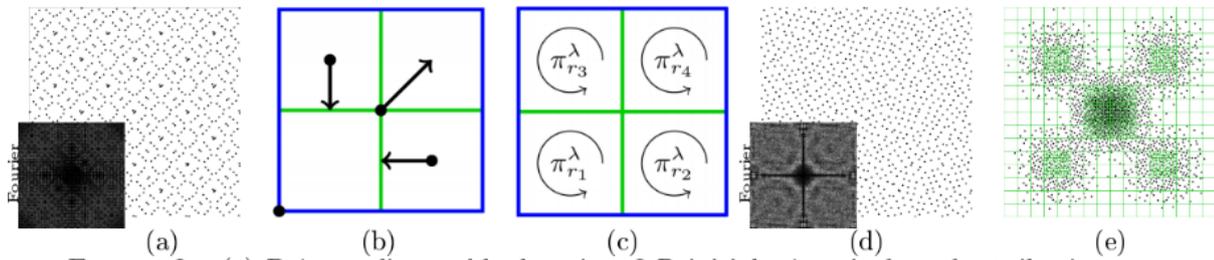


FIGURE 2 – (a) Présente l'ensemble de points 2-D initial, répartis dans des tuiles à un niveau  $l$ . On permute individuellement chaque tuile (b,c) pour obtenir un ensemble qui soit bruit bleu à l'arrivée (d). Une tuile contient toujours le même nombre de points. Pour rajouter des points, on subdivise les tuiles. Cette subdivision peut être faite localement pour faire varier la densité de points (e).