



HAL
open science

Tone Mapping and white balancing for walkthrough

Adrien Gruson

► **To cite this version:**

Adrien Gruson. Tone Mapping and white balancing for walkthrough. Synthèse d'image et réalité virtuelle [cs.GR]. 2011. dumas-00636414

HAL Id: dumas-00636414

<https://dumas.ccsd.cnrs.fr/dumas-00636414>

Submitted on 27 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Master 2 Informatique spécialité Recherche en Informatique : **Rapport de stage**
Université de Rennes 1

Encadrants : Sumanta N. Pattanaik, Rémi Cozot et Kadi Bouatouch

TONE MAPPING AND WHITE BALANCING FOR WALKTHROUGH

Adrien Gruson

School of Electrical Engineering and Computer Science
University of Central Florida, Orlando



le 2 juin 2011

Table des matières

1	Introduction	3
2	Contexte	5
2.1	Système visuel humain	5
2.1.1	L'oeil	5
2.1.2	Adaptation visuelle	6
2.2	High dynamic range	8
3	État de l'art	11
3.1	Illumination globale en temps réel	11
3.2	Tone Mapping	13
3.3	Balance des blancs	15
4	Solutions proposées	17
4.1	Moteur d'illumination globale	18
4.1.1	Moteur 3D expérimental	19
4.1.2	Light Propagation Volumes	20
4.2	Tone mapping dynamique	25
4.3	Balance des blancs	29
5	Résultats	31
5.1	Extension rendu HDR	31
5.2	Opérateur de Tone mapping	32
5.3	Balance des blancs	32
6	Perspectives et Bilan	33
6.1	Perspectives	33
6.1.1	Travaux sur le moteur expérimental	33
6.1.2	Étude d'autres modèles Tone mapping	33
6.1.3	Intégration dans un opérateur commun	34
6.2	Bilan	34

1 Introduction

L'image de synthèse est de plus en plus utilisée dans l'industrie des médias (cinémas, télévision, jeux vidéo). Dans ce cadre, la production d'images de synthèse réalistes représente une part importante dans la génération des blockbusters cinématographiques comme Avatar¹ ou dans le développement de nouveaux jeux comme Crysis 2².

Une grande partie de la recherche dans le domaine de l'image de synthèse 3D s'intéresse au rendu réaliste. On entend par rendu réaliste, un rendu qui nous donnerait la même perception de la scène que si la personne était réellement dans cette scène. Le but des rendus réalistes est d'immerger au mieux le spectateur dans le monde virtuel en le faisant passer pour réel.

La production d'images réalistes fait intervenir plusieurs champs techniques comme la modélisation tridimensionnelle, l'animation ou encore le rendu. Depuis de nombreuses années, la recherche vise à améliorer le rendu réaliste des images de plusieurs façons :

- Une meilleure modélisation des processus physiques intervenant dans la création du rendu.
- L'amélioration de la vitesse de génération de ce type de rendu. En effet, la génération de ce type de rendu est très coûteuse en temps de calcul.
- La prise en compte de la perception visuelle humaine.

On observe également qu'il existe deux types de rendu réaliste :

- Le rendu pré-calculé qui ne présente pas réellement de contrainte temporelle lors de sa génération comme, par exemple, la génération de films de synthèse.
- Le rendu en temps réel ou interactif pour lequel les contraintes temporelles lors de sa génération sont très fortes comme, par exemple, dans les jeux vidéo, la simulation et la réalité virtuelle.

Dans ce champ de recherche, de nombreux chercheurs ont essayé de s'approcher des mécanismes physiologiques complexes de la perception humaine et, en particulier, de celui de la vision.

Par exemple, il est de plus en plus commun de voir des rendus de type HDR (High dynamic range : grande gamme de luminance) dans les jeux vidéo car ils permettent de générer des images qui sont plus correctes physiquement. Cependant, ce type de rendu n'est pas directement affichable par des moniteurs classiques. En effet, pour afficher des rendus de type HDR, il faut d'abord réduire la dynamique de l'image par un opérateur de Tone mapping. L'opérateur de Tone mapping effectue un travail similaire à celui du système visuel humain en s'adaptant à l'éclairage de la scène afin de nous permettre de restituer les détails de l'ensemble de la scène.

1. Film de James Cameron sorti en Décembre 2009

2. Jeu développé par CryTek. sorti en Mars 2011 sous Windows, Playstation 3 et Xbox 360. Jeu utilisant le moteur CryEngine 3. Site web : <http://www.ea.com/crysis-2>

Cependant, cette adaptation a une dimension temporelle qui correspond à un processus biologique complexe. Par exemple, lorsque l'on sort du cinéma en plein journée, notre système visuel met du temps à s'adapter aux nouvelles conditions d'éclairage qui se traduit par une cécité qui disparaît après quelques secondes. Mais lorsque le changement est inverse, par exemple lorsqu'on éteint la lumière avant de se coucher, nous perdons une partie de notre acuité visuelle qui se restaure au bout de plusieurs minutes.

Par ailleurs, le système visuel humain effectue d'autres traitements comme la balance des blancs qui est la faculté du système visuel humain à percevoir le blanc comme blanc quelles que soient les conditions d'éclairage.

C'est donc la combinaison de l'ensemble de ces processus (adaptation à l'éclairage de la scène, balance des blancs, etc.) qui construit notre perception.

Le but de ce stage est de faire converger différentes techniques du système visuel humain en prenant en compte la dimension temporelle de ces processus dans l'optique de recréer virtuellement le travail effectué par notre système visuel humain. Il faudra prendre en compte la cohérence temporelle de la solution. Ceci signifie qu'il faudra éviter les fortes variations non naturelles que pourrait produire la solution.

Organisation du document

Tout d'abord, nous commencerons par expliquer le contexte dans lequel les travaux qui portent sur le Tone mapping et la balance des blancs se sont développés. Après, nous nous intéresserons aux techniques elles-mêmes du point de vue théorique. Ensuite, nous présenterons les différentes solutions que nous avons testées afin de tenter de répondre à la problématique définie pour notre stage. Par exemple, nous présenterons la mise en place d'un moteur d'illumination globale en temps réel. Ce moteur a pour but de fournir les informations de l'éclairage dans une scène dynamique. Enfin, nous présenterons les différentes expérimentations que nous avons menées pour tester ces différentes techniques et leur utilisation en simultanée, puis, nous décrirons les résultats auxquels nous sommes parvenus à ce jour. Pour finir, nous décrirons les perspectives de travail qui seront abordées lors de la suite du stage.

2 Contexte

Ce stage s'effectue dans le domaine de la perception visuelle et de l'image de synthèse. Par conséquent, nous allons d'abord présenter le système visuel humain. En effet, nous allons devoir nous intéresser au système visuel humain car les opérateurs présentés dans la suite du rapport sont basés sur des modèles biologiques. Enfin, nous présenterons le rendu "High dynamic range" (ou HDR) et le rôle qu'il joue dans la génération d'images réalistes.

2.1 Système visuel humain

le système visuel humain est responsable de notre perception visuelle. Il est composé de deux entités importantes qui sont :

- L'œil : Son rôle est de convertir les valeurs physiques de la lumière entrante et de les transformer en signaux électriques afin qu'ils puissent être interprétés par le système nerveux. Cependant, son rôle ne s'arrête pas uniquement à celui de la conversion. En effet, l'œil effectue des traitements en amont de l'information reçue et transmet ses résultats au cerveau. Son rôle a beaucoup été étudié dans les travaux sur la perception visuelle humaine.
- Le cerveau : Son rôle est d'interpréter les informations envoyées par l'œil et de construire notre perception. Malheureusement, du fait de sa complexité, le rôle du cerveau dans la perception n'a été que peu étudié jusqu'à maintenant.

2.1.1 L'œil

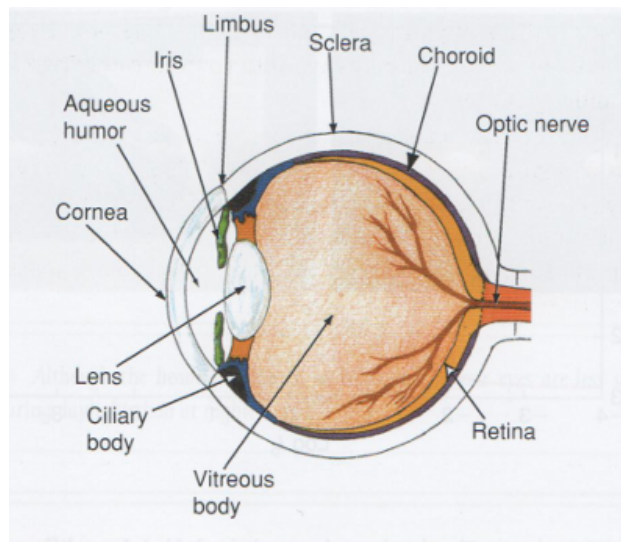


FIGURE 1 – Schéma de l'œil. La lumière incidente entrant par le cristallin se projette sur la rétine.

La rétine est composée de photorécepteurs dont le rôle est de convertir les valeurs physiques de la lumière en signaux électriques. Il existe deux types de photorécepteurs :

- les bâtonnets : ils sont très sensibles à la lumière et responsables de notre acuité visuelle scotopique ($10^{-1}cd/m^2$ à $10^{-6}cd/m^2$). Les bâtonnets, qui sont achromatiques, sont très vite saturés en journée.
- les cônes : ils sont responsables de notre faculté à voir les couleurs et de notre acuité visuelle photopique ($10^1cd/m^2$ à $10^8cd/m^2$). Il existe trois types de cônes (Large, Medium et Short). Les cônes sont moins sensibles à la lumière que les bâtonnets.

2.1.2 Adaptation visuelle

Le système visuel humain s'adapte aux conditions d'illumination. Par exemple, les phares d'une voiture ont une puissance qui ne varie pas. Cependant, ces phares ne se verront presque pas en journée alors qu'ils nous servent pour conduire la nuit. Des expériences ont été faites pour le montrer par rapport à une luminance de fond et avec un petit stimulus. Par exemple, une expérience nommée "Threshold-versus-intensity" (TVI) cherche à trouver quel est l'incrément le plus faible par rapport à la luminance de fond pour percevoir ce stimulus (figure 2). La luminance de fond autour du stimulus est aussi appelée la luminance d'adaptation.

La figure 2, ci-dessous, présente les courbes TVI en fonction de la luminance d'adaptation, soit la courbe de l'expérience pour les deux types de photorécepteurs.

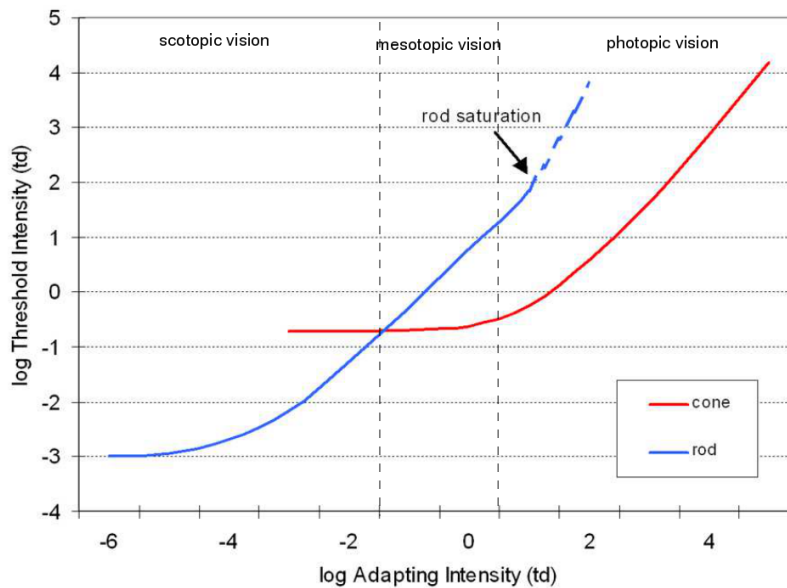


FIGURE 2 – On remarque bien que les bâtonnets arrivent rapidement à saturation. Image tirée de l'article [10]

D'autres expériences mettent en évidence le processus d'adaptation des photorécepteurs. Elles montrent que si les photorécepteurs sont exposés à une intensité de fond forte pendant un certain temps, alors ils ne restent pas saturés. En effet, les photorécepteurs restaurent graduellement leur sensibilité par rapport à la nouvelle luminance de fond. On parle de réponse électrique relative des photorécepteurs, car ceux-ci envoient des signaux électriques à différentes fréquences. Plus un

récepteur est saturé, plus l'intervalle entre deux signaux électriques est court. Ces réponses ont été mesurées directement dans la rétine de rats grâce à des électrodes. On peut modéliser la réponse électrique relative aux photorécepteurs avec l'équation de Naka-Rushton :

$$\frac{R}{R_{max}} = \frac{I^n}{I^n + \sigma_b^n} \quad (1)$$

Dans cette formule $\frac{R}{R_{max}}$ correspond à la réponse électrique relative aux photorécepteurs avec R_{max} la réponse maximale des photorécepteurs et R la réponse du photorécepteur. Si cette réponse relative est égale à 1, cela signifie que les photorécepteurs sont saturés. σ_b est le paramètre de demi saturation, c'est-à-dire la valeur d'intensité pour que la réponse électrique des photorécepteurs soit égale à 0.5. Enfin, I correspond à l'intensité du stimulus et n correspond à une constante entre 0.7 et 1.0 [20].

Par ailleurs, σ_b peut être relié à la luminance d'adaptation. Cette relation empirique a été proposée par Xie et al. [20] :

$$\sigma = I_A^\alpha \times \beta \quad (2)$$

Dans l'équation 2, I_A correspond à la luminance d'adaptation, α et β sont des constantes. Cette luminance d'adaptation est très importante, car c'est elle qui est responsable de l'adaptation visuelle.

D'autre part, l'adaptation visuelle humaine n'est pas un processus instantané. En effet, certains auteurs [10] modélisent cette dimension temporelle en effectuant un filtrage temporel sur la luminance d'adaptation. Ils distinguent deux types d'adaptation :

- "dark adaptation" : cette adaptation correspond à une chute de la luminance "globale" dans la scène, par exemple, quand on éteint une lampe. L'effet perçu par cette adaptation est une perte de perception de la scène. Cet effet s'estompe lentement dans le temps.
- "light adaptation" : cette adaptation correspond à une augmentation de la luminance "globale" dans la scène, par exemple, quand on allume la lumière en se levant le matin. L'effet perçu par cette adaptation est un aveuglement temporaire. Cet aveuglement disparaît rapidement dans le temps.

Cependant, la modélisation de cette adaptation n'est pas facile car les cônes et les bâtonnets n'ont pas le même comportement en fonction des différents types d'adaptation. Par exemple, pour la "dark adaptation" (figure 3), plus le niveau d'adaptation a été élevé avant le changement d'éclairage, plus on remarque une adaptation en deux étapes. Cela est dû aux cônes qui adaptent plus rapidement leur sensibilité mais atteignent rapidement leurs limites.

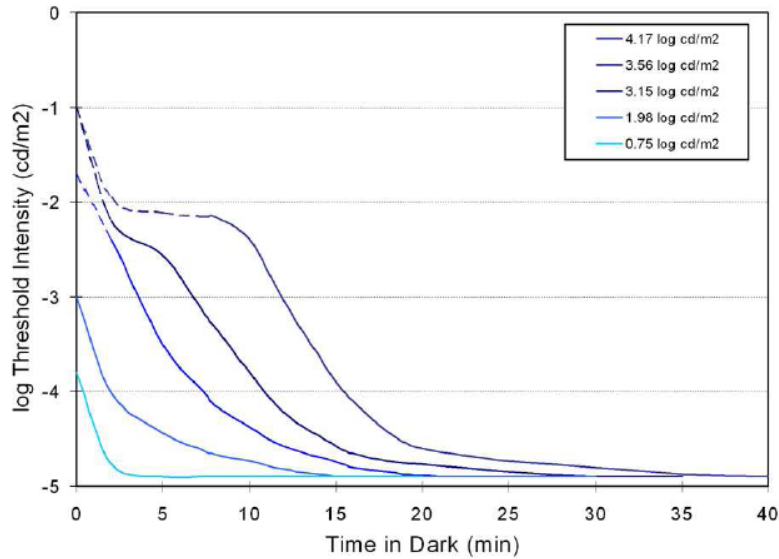


FIGURE 3 – Courbe montrant la dimension temporelle de l'adaptation pour une "dark adaptation" en fonction de différentes luminances de pré-adaptation. Image tirée de l'article [10]

2.2 High dynamic range

Les images "High Dynamic Range" (ou HDR) sont des images spéciales. Les images HDR ont pour but de pouvoir stocker toutes les valeurs de luminance de la scène. En effet, dans une scène réelle, les différentes valeurs de luminance peuvent fortement varier à partir d'un même point de vue. Par exemple, dans une scène réelle, on trouvera une luminance de 10^4 cd/m^2 pour le soleil alors qu'une zone d'ombre dans l'image n'aura qu'une luminance de 0.1 cd/m^2 (figure 4). Donc, la principale différence entre des images HDR et des images normales (nommées aussi LDR pour "Low Dynamic Range") réside dans la quantité de valeurs que l'on peut stocker.

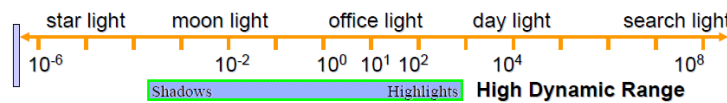


FIGURE 4 – Une scène réelle peut être composée de valeurs de luminance très éloignées.

Ce type d'image a tout d'abord été développé pour le domaine de la photographie. Par exemple, il est possible de produire une image HDR en utilisant plusieurs images LDR prises à partir du même point de vue. En effet, en réglant les temps d'obscurité du diaphragme d'un appareil photo on peut contrôler le montant d'énergie reçu sur les capteurs de l'appareil. Donc, en réglant la quantité d'énergie entrant dans l'appareil, il est possible de capturer l'ensemble de la dynamique de la scène.

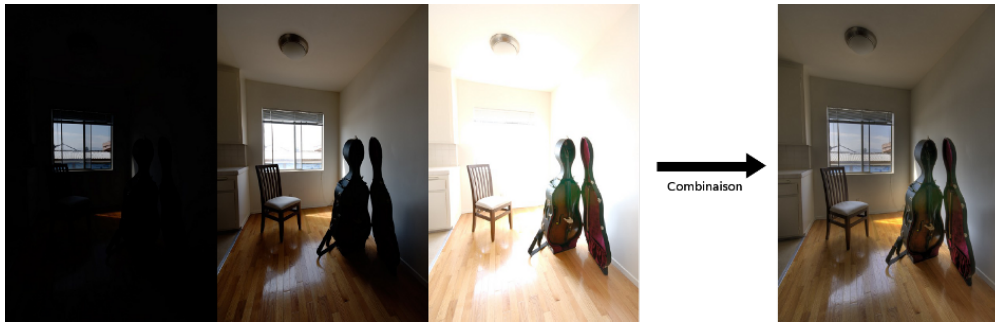


FIGURE 5 – Images de gauche : plusieurs images produites avec des temps d’expositions différents. On remarque que chacune de ces images ne nous donne qu’une partie des informations de la scène. Image de droite : en combinant les différentes images LDR, il est possible de produire une image HDR. L’image affichée a subit un traitement pour qu’elle puisse être affichée.

Cependant, avec cette technique, il est difficile d’avoir les valeurs absolues des différentes luminances de la scène à cause du calibrage inconnu des différents capteurs de l’appareil photo. Cependant, certains appareils peuvent produire des images HDR calibrées comme le montre l’image de la figure 5³ :



FIGURE 6 – Exemple d’une scène réelle prise avec un appareil photographique pouvant produire directement des images HDR. Les valeurs de luminance affichées varient de 1 à $10^5 cd/m^2$

3. image disponible à l’adresse suivante : <http://white.stanford.edu/~brian/hdr/hdr.html>

Si le calibrage est inconnu, alors il faut appliquer d'abord l'équation 6 à l'image HDR produite. Cette formule part du fait que l'on peut connaître le type de scène (nuit, jour, soir) en fonction des différentes valeurs de la luminance minimale (L_{min}), la luminance maximale (L_{max}) et la luminance moyenne (L_{av}).

$$L_{av} = \exp\left(\frac{1}{N} \sum_{i=1}^N \log(L_w(x, y) + \epsilon)\right) \quad (3)$$

$$f = \frac{2 \log_2 L_{av} - \log_2 L_{min} - \log_2 L_{max}}{\log_2 L_{max} - \log_2 L_{min}} \quad (4)$$

$$\alpha = 0.18 \times 4^f \quad (5)$$

$$L'_w(x, y) = \frac{\alpha}{L_{av}} L_w(x, y) \quad (6)$$

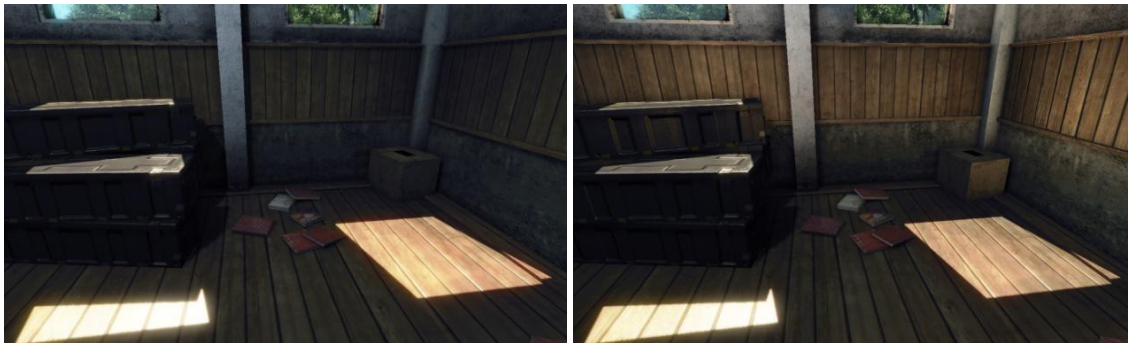
avec $L'_w(x, y)$ la luminance calibrée et $L_w(x, y)$ la luminance non calibrée.

Cependant, dans notre problématique, nous allons tenter de simuler directement les valeurs de l'éclairage. Ceci signifie que nous n'aurons pas besoin d'effectuer un calibrage avant d'appliquer nos opérateurs simulant le comportement du système visuel humain.

3 État de l'art

3.1 Illumination globale en temps réel

L'essor des méthodes d'illumination globale en temps réel a été rendu possible grâce aux évolutions des cartes graphiques (ou GPU⁴). En effet, la puissance de celles-ci a considérablement augmenté ces dernières années. De plus, les constructeurs de cartes graphiques ont donné la possibilité aux développeurs de pouvoir tirer partie de cette puissance en leur permettant d'exécuter des codes directement sur la carte graphique. Cela a été possible avec l'avènement des shaders qui sont des codes particuliers compilés et exécutés directement sur le GPU.



(a) Rendu sans illumination globale. Utilisation d'un terme ambiant pour modéliser l'éclairage indirecte.

(b) Rendu avec illumination globale.

FIGURE 7 – Différence entre un rendu avec ou sans illumination globale. Image tirée du rapport de CryTek [5]

Cependant, pour qu'un algorithme puisse être implémenté sous GPU, il faut qu'il soit fortement parallélisable. De plus, les limites matérielles des cartes graphiques (mémoire disponible, etc.) et leur architecture particulière rendent plus difficile le développement sur celles-ci.

Il existe beaucoup de techniques de calcul d'illumination globale en temps réel. Cependant, on peut les classer rapidement en plusieurs catégories de techniques :

Techniques classiques

Initialement, ces techniques ont été développées sur CPU⁵. Par exemple, elles sont basées sur les techniques de lancer de rayon ou de radiosité. Beaucoup de recherches ont été faites pour permettre à ces techniques de s'exécuter sur GPU afin accélérer les calculs.

Techniques basées sur des données pré-calculées

Elles utilisent des données pré-calculées sur la scène pour pouvoir accélérer les calculs. Par exemple, certaines d'entre elles se basent sur un pré-calcul des transferts de radiosité dans la scène sous forme d'harmoniques sphériques.

4. GPU : Graphics processing unit

5. CPU : Central processing unit

Techniques basées dans le domaine image

Elles ont connu un essor récent grâce à la démocratisation des solutions basées sur GPU. Ces techniques travaillent uniquement dans l'espace image, c'est-à-dire uniquement à partir du point de vue de la caméra. Le problème de ce genre de technique est que leur application conduit à restreindre le point de vue de manière très importante et, par conséquent, à faire perdre beaucoup d'informations sur la scène.

Techniques de type "Instant radiosity"

Elles sont basées sur le fait que l'on peut approcher l'illumination globale de la scène avec un ensemble de lampes virtuelles. Les lampes virtuelles sont des lampes créées directement sur le GPU. Elles permettent d'effectuer une interpolation de l'illumination sur la scène et de réduire la complexité de l'éclairage.

Techniques basées sur représentation en voxels / Techniques discrètes

Elles sont basées sur une discrétisation de la scène pour simplifier les différents calculs inhérents aux problèmes d'illumination globale. Par exemple, ce type d'approche est beaucoup utilisée pour le rendu des milieux participatifs.

Cependant, il est important de préciser que certaines méthodes peuvent appartenir à plusieurs catégories. Par exemple, la méthode "Voxel-based global illumination" [17] qui consiste à faire une voxelisation de la scène et à utiliser une méthode à base de lancer de rayon pour calculer l'éclairage global. Ou encore, la méthode [6] que l'on a choisie pour notre stage qui utilise les lampes virtuelles pour générer l'information de l'éclairage dans la scène mais qui utilise également des techniques basées sur une représentation en voxels pour calculer l'illumination globale.

3.2 Tone Mapping

Les images HDR, avec leur grande dynamique, ne peuvent pas être affichées directement sur un moniteur classique. En effet, un moniteur classique ne peut afficher que des luminances allant, par exemple, de 0 à 300 cd/m^2 . Ceci ne convient pas à l'image HDR qui, elle, peut contenir des luminances allant de 0 à 10^6 cd/m^2 . Ce constat montre qu'il est nécessaire de trouver une fonction qui permette de compresser la dynamique de l'image HDR afin de l'afficher sur un moniteur LDR.

L'opérateur de Tone Mapping a pour but de trouver une fonction pour convertir une image HDR en une image LDR. Cependant, trouver ce type d'opérateur n'est pas trivial car il doit minimiser l'erreur de perception entre l'image HDR et l'image LDR après l'application de l'opérateur de Tone Mapping (figure 13 et 9).

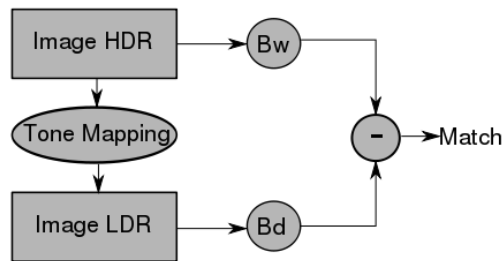


FIGURE 8 – B_w correspond à la luminosité perçue par l'œil directement. B_d correspond à la luminosité perçue par l'œil sur l'écran. L'opérateur de Tone Mapping cherche à avoir la même luminosité pour les deux images.

Dans leur fonctionnement, les opérateurs de Tone mapping jouent le rôle de l'adaptation visuelle humaine. En effet, leur but est de transformer les valeurs physiques de l'éclairage et de nous donner une perception correcte de celles-ci à travers le système d'affichage (écran). Cependant, ce processus n'est pas le seul qui intervient dans la formation de notre perception. La plupart des opérateurs de Tone mapping se compose de deux modèles (figure 9) :

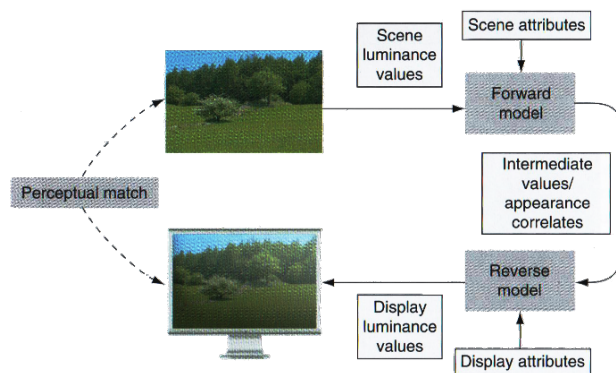


FIGURE 9 – En haut, se trouve l'image HDR, celle-ci contient les valeurs de luminance de la scène. En bas, se trouve l'image LDR. L'opérateur de Tone Mapping qui va compresser l'image HDR doit prendre en compte les spécificités de la scène (Forward model) et aussi les conditions de visualisation de l'image (Reverse model).

Le modèle direct (Forward model) effectue une simple compression sous la forme d'une sigmoïde ; il s'inspire du modèle proposé par Naka-Rushton. Soit L_a la luminance d'adaptation de l'image HDR, L_w la luminance de l'image HDR et n une constante :

$$V(x, y) = \frac{L_w^n(x, y)}{L_w^n(x, y) + L_a^n(x, y)} \quad (7)$$

le modèle inverse (Reverse model) effectue une mise à l'échelle du résultat du Forward model ($V(x,y)$) pour qu'il puisse correspondre à la luminance du moniteur. Soit $L_{d,mean}$ la médiane de la luminance du moniteur, m une constante du moniteur et L_d la luminance affichable :

$$L_d(x, y) = \left(\frac{V(x, y)L_{d,mean}^m}{1 - V(x, y)} \right)^{\frac{1}{m}} \quad (8)$$

Pour calculer la luminance d'adaptation de l'image HDR, les opérateurs de Tone Mapping se décomposent en plusieurs groupes :

- Opérateur global : ce type d'opérateur calcule une seule luminance d'adaptation pour toute l'image.
- Opérateur local : ce type d'opérateur calcule la luminance d'adaptation pour chaque pixel de l'image en fonction de son voisinage.

D'autre part, exception faite des opérateurs de Tone mapping travaillant dans l'espace colorimétrique HSV, il est généralement accepté qu'un opérateur de Tone mapping ne travaille que sur la luminance⁶. Cependant, pour revenir à l'espace colorimétrique RGB, il faut effectuer la transformation suivante :

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} L_d \left(\frac{R_w}{L_w} \right)^s \\ L_d \left(\frac{G_w}{L_w} \right)^s \\ L_d \left(\frac{B_w}{L_w} \right)^s \end{bmatrix} \quad (9)$$

avec L_d la luminance résultante de l'opération de Tone mapping, (L_w, R_w, G_w, B_w) la luminance et les valeurs colorimétriques de l'image HDR et $s \in [0, 1]$ le paramètre de saturation.

6. Tirée du livre "High dynamic range Imaging" 2nd édition [4], page 268

3.3 Balance des blancs

La balance des blancs consiste à prendre en compte le travail du système visuel humain relatif à l'adaptation chromatique. Une adaptation chromatique est une adaptation qui change l'espace de couleurs et permet de le transformer dans un autre espace de couleurs. Ceci signifie que cette adaptation touche uniquement la manière dont on perçoit les couleurs à partir des valeurs physiques de la lumière. En effet, le système visuel humain ne prend pas en compte uniquement la distribution spectrale de la luminance incidente à l'œil mais transforme celle-ci pour construire notre perception des couleurs.

Par exemple, une pièce blanche éclairée d'une lampe orange sera perçue blanche alors que le résultat d'un algorithme d'illumination globale nous donnera une scène complètement orange [13] (illustrée par l'image 10).



(a) Image sans balance des blancs effectuée (b) Image avec prise en compte de la balance des blancs avec la technique Grey World (c) Image avec prise en compte de la balance des blancs avec la technique proposée dans [16].

FIGURE 10 – Images pour illustrer le problème de balance des blancs. Ces images sont tirées de l'article de Cozot et al. [16]

Les algorithmes de balance des blancs essaient d'estimer l'illuminant de référence pour l'annuler grâce à une adaptation chromatique. La plupart de ces algorithmes suivent un processus en deux étapes :

- rechercher dans la scène l'illuminant de référence correspondant à notre blanc de référence. Cette phase est la plus complexe à réaliser.
- appliquer une adaptation chromatique avec l'illuminant de référence.

La plupart des techniques de balance des blancs n'effectuent pas la recherche de l'illuminant de référence de la même manière. Cependant, il existe deux types de contextes d'estimation :

- Contexte 2D : correspond à une image prise par un appareil photographique. Ce type de technique est uniquement utilisée dans le domaine de l'image.
- Contexte 3D : correspond à une connaissance du modèle complet de la scène. C'est ce cas que nous utilisons car nous avons recours à un moteur d'illumination globale pour produire nos images.

L'adaptation chromatique a pour but de simuler le travail du système visuel humain à partir d'un illuminant de référence. Pour ce faire, il faut passer dans l'espace LMS qui est l'espace de couleur des cônes. Cependant, il existe plusieurs matrices de transformation qui permettent de passer de l'espace colorimétrie XYZ dans l'espace de cônes LMS, les trois principales étant :

- La matrice de von Kries (1902) [18]
- La matrice de Bradford [7]
- La matrice utilisée dans CIECAM02 qui se nomme M_{CAT02} [12]

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = M_{CAT02} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (10)$$

Ensuite, l'hypothèse de von Kries stipule que l'adaptation chromatique est faite indépendamment des différents types de cônes.

$$\begin{bmatrix} L_a \\ M_a \\ S_a \end{bmatrix} = \begin{bmatrix} \frac{1}{L_w} & 0 & 0 \\ 0 & \frac{1}{M_w} & 0 \\ 0 & 0 & \frac{1}{S_w} \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (11)$$

Cette formule montre comment calculer les nouvelles valeurs colorimétriques en prenant en compte l'adaptation chromatique. Les valeurs (L_w, M_w, S_w) représentent l'illuminant de référence et les valeurs (L, M, W) représentent l'image non adaptée. Après l'adaptation chromatique, on trouve les valeurs (L_a, M_a, S_a) qui sont maintenant adaptées à notre illuminant de référence. Enfin, pour afficher le résultat corrigé chromatiquement, il nous faut revenir dans l'espace colorimétrie XYZ.

Enfin, l'opérateur de balance des blancs devra avoir une cohérence temporelle. C'est-à-dire que des changements brusques dans l'illuminant de référence ne doivent pas impacter la perception finale du rendu. En effet, il se peut qu'il y ait des apparitions d'artefacts comme des changements brutaux au niveau des couleurs rendues.

4 Solutions proposées

Pour le développement des solutions liées à la problématique du stage, nous avons procédé de la manière suivante :

- Implémentation d’une méthode d’illumination globale. En effet, la première étape était de générer des données HDR pour ensuite développer des opérateurs de Tone mapping et de balance des blancs.
- Implémentation des opérateurs de Tone mapping. Le but était de pouvoir produire des images LDR que l’on pourra utiliser pour effectuer la balance des blancs.
- Implémentation des opérateurs de balance des blancs.

En effet, le pipeline de rendu que l’on veut mettre en place est le suivant :

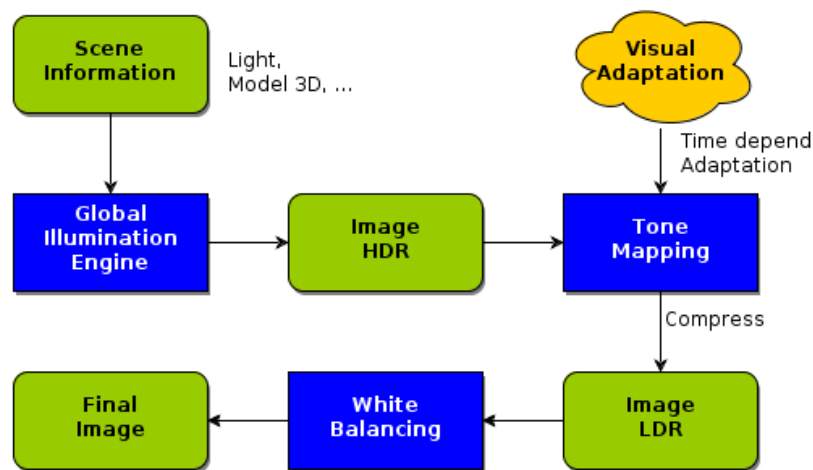


FIGURE 11 – Le pipeline général proposé. Nous commençons par créer une image HDR avec le moteur d’illumination globale. Ensuite, nous réduisons la dynamique de cette image HDR avec des opérateurs de Tone mapping. Enfin, nous effectuons une correction chromatique de l’image LDR avec une technique de balance des blancs.

C’est pour cette raison que l’on commencera par présenter le travail sur le moteur d’illumination globale. Ensuite nous passerons à l’opérateur de Tone mapping. Enfin, nous examinerons les opérateurs de balance des blancs.

4.1 Moteur d'illumination globale

Pour générer des images HDR en temps réel, il a fallu développer un moteur d'illumination globale en temps réel. Pour ce faire, nous avons décidé d'implémenter la technique proposée par CryTek : "Cascaded Light Propagation Volumes for Indirect Illumination" [6]. Ce choix a été motivé par plusieurs raisons :

- Cette technique calcule en temps réel l'illumination globale sans aucune donnée pré-calculée. Cela nous donnait la possibilité d'effectuer des rendus pour des scènes entièrement dynamiques.
- Cette technique est très peu coûteuse en temps de calcul du fait des nombreuses approximations effectuées dans le calcul de l'illumination. Mais elle donne des résultats plausibles.

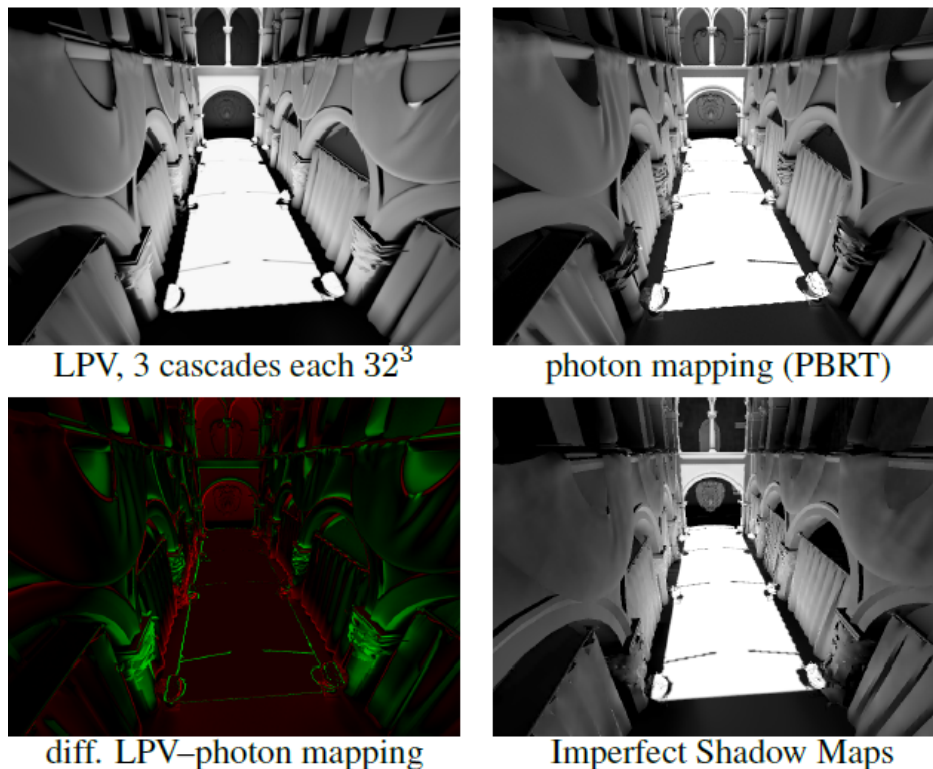


FIGURE 12 – Voici la comparaison entre la technique de CryTek (temps de calcul 1/60 s), un rendu à base de photon mapping (temps de calcul 2700 s) et une autre technique d'illumination globale en temps réel : "Imperfect Shadow Maps"

Toutefois, avant d'implémenter cette technique, nous avons mis en place un moteur 3D expérimental dans le but de faciliter les implémentations futures sur GPU. Dans la suite de cette partie, nous expliquerons les apports de ce moteur 3D expérimental. Puis, nous décrirons plus en détail la méthode de CryTek.

4.1.1 Moteur 3D expérimental

Pour mener les expérimentations, il a fallu d'abord mettre en place une expérimentation du moteur 3D dont le but est de faciliter la mise en œuvre des différentes techniques rencontrées pendant notre stage. En effet, la plupart des techniques actuelles s'appuient sur la possibilité de programmer le pipeline graphique grâce aux shaders.

Par ailleurs, nous avons fait un choix technologique qui nous a conduit à implémenter ce moteur avec une version supérieure ou égal à la version 3 d'OpenGL. Nous avons choisi cette version car, à partir de celle-ci, l'API OpenGL a fortement changé. En effet, beaucoup de fonctionnalités proposées par OpenGL ont été dépréciées. Ce changement a été effectué dans l'optique de privilégier l'implémentation par les shaders. Cependant, beaucoup de méthodes comme la gestion des matrices sont maintenant à la charge du développeur.

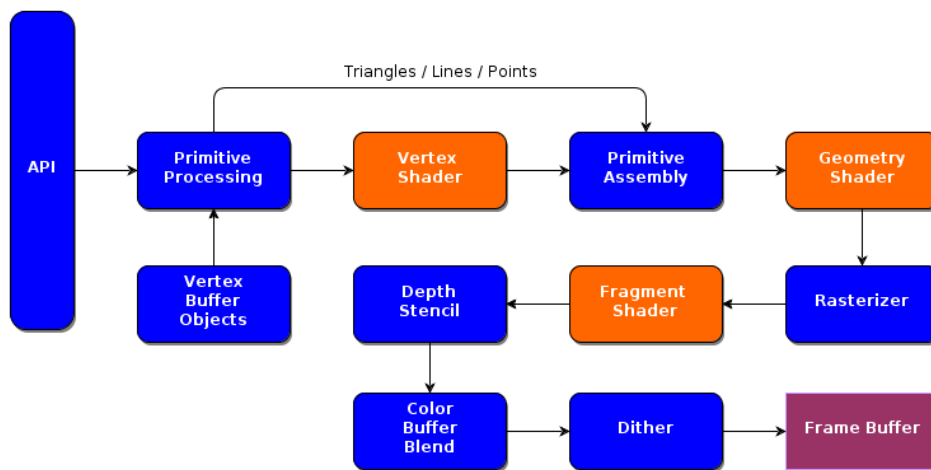


FIGURE 13 – Le pipeline OpenGL v3+. Les rectangles oranges indiquent toutes les possibilités que le développeur a à sa disposition pour programmer le GPU.

Par ailleurs, dans l'optique de faciliter le développement de techniques à base de shader, nous proposons une syntaxe de déclaration du shader sous forme XML. En effet, avant de pouvoir utiliser un shader, il faut d'abord le charger sur la carte graphique et le configurer correctement. Le fichier XML sert à savoir comment charger le shader mais surtout à configurer automatiquement les valeurs d'entrées et de sorties du shader. Ce procédé permet de réduire au minimum le temps et la complexité du travail du développeur qui utilise les shaders dans son application.

De manière complémentaire, nous utilisons également la bibliothèque Assimp pour charger les modèles 3D et FreeImage pour charger les images. Nous avons aussi mis en place un graphe de scène basique pour représenter de façon hiérarchique notre scène.

Ce moteur graphique se nomme : "Amaterasu3D" ; il est consultable à l'adresse suivante : <https://github.com/beltegeuse/Amaterasu3D>. Il est mis à disposition sous licence GPL version 2+.

4.1.2 Light Propagation Volumes

CryTek propose une solution pour calculer une approximation de l'illumination globale de façon rapide. Leur but est d'avoir une méthode de calcul d'illumination globale qui puisse effectuer l'ensemble des calculs en temps réel. D'autre part, ils veulent que cette technique puisse fonctionner dans des environnements entièrement dynamiques qui ne permettent pas d'utiliser des informations pré-calculées. Pour résumer, en développant cette technique, leur objectif est de :

- n'utiliser aucune donnée pré-calculée. Ceci revient, pour chaque image, à ne pas réutiliser de données existantes.
- ne calculer qu'un seul rebond indirect de lumière dans la scène mais en prenant en compte son occultation.
- avoir une technique rapide car cette solution devra tourner sur des consoles avec un budget temps limité.



FIGURE 14 – Schéma donnant l'ensemble du pipeline dans le CryTek engine 3. On remarque que l'illumination globale doit être calculée en seulement 3.3 ms.

Cependant, pour réussir à atteindre leurs buts, les auteurs ne veulent qu'un résultat plausible et non précis. De plus, les auteurs ne veulent prendre en compte que des scènes ayant un nombre de réflecteurs diffus prépondérants. Cela signifie que pour des scènes ayant beaucoup d'objets spéculaires, les résultats de cette technique seront faux.

Par conséquent, les auteurs proposent d'utiliser des harmoniques sphériques de bas ordre (4) et de les stocker sous forme de volume. Cependant, leur utilisation ne permettra de calculer que l'éclairage indirect de basse fréquence. D'ailleurs, pour réduire ce problème, les auteurs proposent l'utilisation de la technique "Screen space ambient occlusion" [11] dans le but d'ajouter des ombres haute fréquence au rendu final.

Dans cette technique, les auteurs utilisent plusieurs volumes avec une représentation par des harmoniques sphériques pour calculer l'éclairage indirect :

- Light propagation volume (ou LPV) : Ce volume contiendra les informations sur l'éclairage. Ainsi, pour chaque composant colorimétrique (R,V,B), nous devons avoir un volume.
- Geometry volume : Ce volume contiendra les informations liées à la scène. Il servira à effectuer l'occultation dans la phase de propagation.

Les principales étapes de cette technique sont :

- Initialisation des LPV : Pour initialiser la grille contenant l'éclairage (ou LPV), il nous faut injecter les contributions de chaque lampe dans la scène.
- Injection des géométries de la scène : Pour initialiser la grille contenant les informations sur la scène, il nous faut injecter les géométries visibles depuis les lampes et la caméra.
- Propagation de l'éclairage : Ensuite, nous utilisons la propagation de l'éclairage se trouvant dans les LPV proposés par les auteurs. Pour ce faire, les auteurs utilisent une propagation de type DOM⁷. De plus, nous utilisons la grille contenant les informations sur la scène pour effectuer du blocage flou⁸ et arrêter la propagation.
- Rendu de la scène : Nous utilisons les données disponibles dans le LPV ayant subit la propagation pour calculer l'éclairage sur chaque objet visible depuis la caméra.

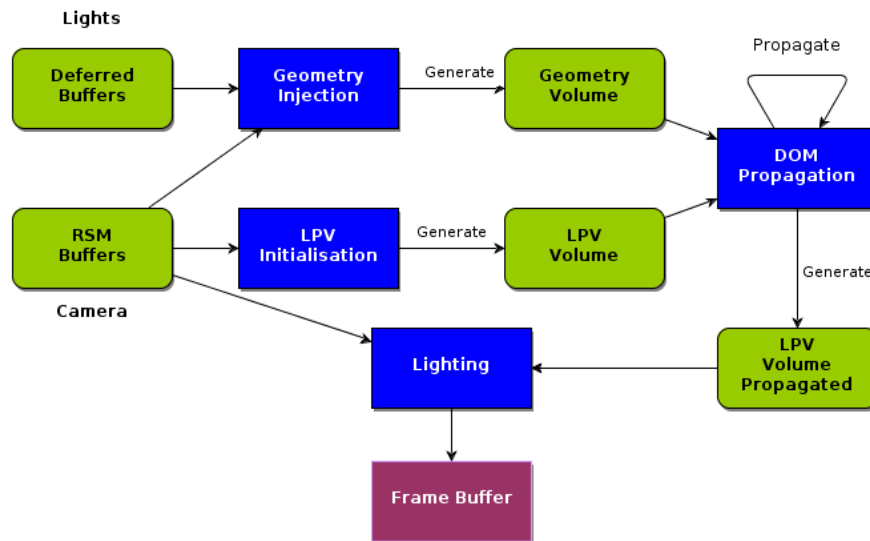


FIGURE 15 – Schéma récapitulant les différentes phases de la technique de CryTek. Les auteurs utilisent les techniques "Reflective shadow maps" (ou RSM) [2] et de "Deferred shading"¹⁰. dans le but de décorréler la complexité de la scène et celle de l'éclairage

7. DOM : discrete ordinate method. Ce sont des méthodes basées sur des discrétisations des différentes directions possibles de propagation et du volume sous forme de voxels.

8. On entend par blocage flou la modélisation de celui-ci par des probabilités

Initialisation des LPV

Pour chaque lampe, nous utilisons les informations mises à disposition dans les buffers générés par la technique RSM pour construire un ensemble de lampes virtuelles. Ensuite, nous convertissons chaque lampe virtuelle en représentation d'harmonique sphérique et nous les injectons dans les LPV en ajoutant leur contribution respective. De plus, pour éviter des problèmes comme l'auto-ombrage, les auteurs proposent de déplacer virtuellement la lampe virtuelle d'une distance de la moitié de la taille d'une cellule du LPV dans la direction de la normale de la surface.

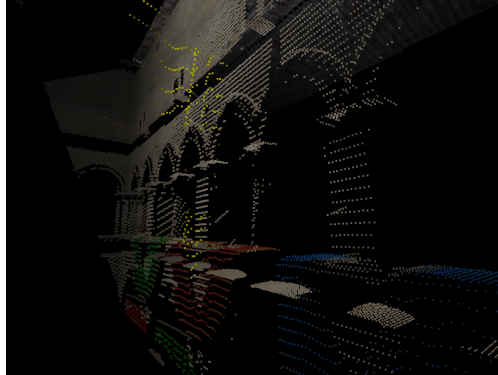


FIGURE 16 – Exemple de visualisation de la génération de lampes virtuelles à partir d'une lampe directionnelle. Pour cela, nous utilisons un échantillonnage constant sur le buffer flux du RSM.

Injection des géométries de la scène

A partir du point de vue des lampes et de la caméra, les auteurs proposent de calculer la probabilité de blocage (équation 12) pour chaque surface vue. On transforme ensuite cette valeur en représentation d'harmoniques sphérique puis on les injecte dans le volume contenant les informations de la scène. On utilise des buffers séparés pour chaque point de vue puis on fusionne tous les volumes produits en utilisant l'opération max des harmoniques sphériques pour éviter d'injecter deux fois la même surface.

$$B(w) = A_s s^{-2} \langle n_s | w \rangle_+ \quad (12)$$

avec A_s l'aire de la surface vue à travers le texel, s la dimension d'une cellule du LPV, n_s la normale de la surface et w la direction de la vue. $B(w)$ est la représentation en harmonique sphérique du blocage flou. Notez que l'opérateur $\langle . | . \rangle_+$ est un produit scalaire sans valeurs négatives.

Cependant, si les informations liées à la scène sont trop peu importantes, les auteurs proposent les solutions suivantes :

- Effectuer du "depth-peeling" à partir de la caméra pour extraire plus d'informations sur la scène (très coûteux).
- Utiliser les anciennes valeurs de ce volume pour donner plus d'informations sur la scène.

10. On entend par "Deferred shading" la création de plusieurs buffers (Position ou profondeur, normales, informations matériaux) depuis le point de vue de la caméra.

Par ailleurs, les auteurs suggèrent de ne pas injecter les géométries plus petites que la taille d’une cellule de ce volume car celles-ci peuvent provoquer l’apparition d’artefacts. Cependant, dans notre implémentation, nous avons choisi de les injecter car sinon cela signifierait devoir régénérer les différentes informations depuis les sources de lumière et de la caméra.

Propagation de l’éclairage

Ensuite, il nous faut calculer l’éclairage indirect. Pour ce faire, les auteurs proposent d’utiliser les LPV et d’effectuer une propagation de type DOM pour diffuser l’éclairage dans la scène. Ils ont choisi cette technique car celle-ci est très facilement parallélisable sur GPU.

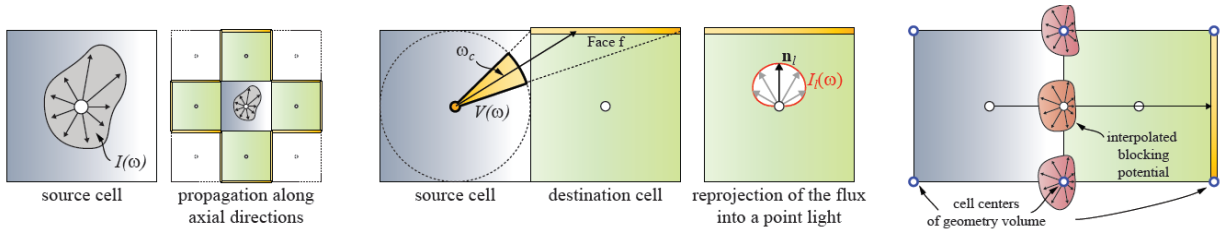


FIGURE 17 – Gauche : Chaque cellule du LPV contient les intensités directionnelles utilisées pour propager son intensité aux six voisins de cette cellule. Centre : Pour chaque cellule voisine, on projette l’intensité de la cellule source sur chaque face visible de la cellule de destination en passant par la création d’une lampe virtuelle. Ensuite, on projette la lampe virtuelle créée dans la cellule de destination en la transformant en représentation d’harmonique sphérique. Droite : Pour chaque calcul de projection, nous utilisons le volume contenant les géométries pour effectuer du blocage flou.

Rendu de la scène

Pour effectuer le rendu final, nous utilisons les informations contenues dans les LPV pour connaître l’éclairage local sur chaque objet visible depuis la caméra. Pour calculer cet éclairage, les auteurs utilisent une interpolation trilinéaire des informations contenues dans la LPV. Ensuite, ils convertissent cette information interpolée en lampe virtuelle et effectuent l’éclairage. Cependant, ce type de méthode peut donner lieu à des erreurs d’interpolation (illustrée par la figure 18).



FIGURE 18 – Gauche : Utilisation d'un filtrage trilineaire. Le problème vient du fait que le toit est plus fin que la taille d'une cellule du LPV. Donc lors de l'interpolation, on prend en compte l'éclairage sur le dessus du toit ce qui produit du faux éclairage. Droite : CryTek propose une technique d'interpolation permettant de contrer ce problème. Pour ce faire, ils déplacent virtuellement l'éclairage dans la direction opposée au gradient par rapport à la normale du point.

Version cascades

Dans la technique de CryTek, les auteurs proposent d'améliorer les résultats en utilisant plusieurs niveaux de LPV de différentes tailles positionnés autour de la caméra. Le but recherché est d'avoir plus de précision sur l'éclairage pour les objets proches de la caméra et moins pour les objets éloignés.

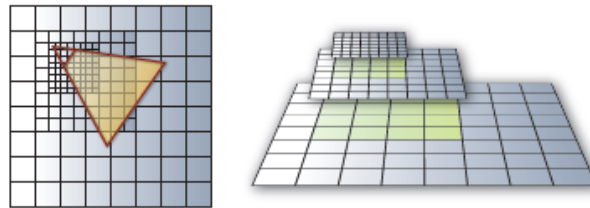


FIGURE 19 – Les différents niveaux de LPV sont placés de façon à maximiser la couverture de la caméra. D'autre part, les différents niveaux sont imbriqués ensemble.

Par ailleurs, pour éviter toute apparition d'artefact lors du déplacement de la caméra, les différentes grilles ne se déplacent que d'une distance multiple de la taille d'une cellule de chaque grille. Donc, nous effectuons un déplacement de la grille lorsque la caméra s'est déplacée de plus d'une taille de cellule.

Mais encore, pour éviter un surcoût trop important dans les phases d'injection, nous avons choisi d'utiliser les "geometry shaders" qui permettent, en une seule injection, d'effectuer l'injection sur les différentes grilles. Pour ce faire, nous utilisons la possibilité que nous donne les "geometry shaders" de créer de nouveaux sommets librement. En effet, nous avons choisi de mettre les différents niveaux sur une même image dans le but de réduire le nombre de MRT¹¹ activés en même temps.

11. MRT : Multiple Render Targets. C'est la possibilité qu'offrent les carte graphique d'écrire dans plusieurs buffers à la fois.

4.2 Tone mapping dynamique

Une des problématiques de ce stage est de s'intéresser à la dimension temporelle des opérateurs de Tone mapping. Il est intéressant de montrer la différence entre un opérateur prenant en compte la dimension temporelle et ce même opérateur ne prenant pas en compte cette dimension. D'autre part, il serait intéressant d'étudier plusieurs types d'opérateur comme :

- des opérateurs de Tone mapping global et, dans ce cas, nous calculerons pour chaque image la nouvelle luminance d'adaptation.
- des opérateurs de Tone mapping local et, dans ce cas, nous utiliserons un filtrage pour calculer les nouvelles luminances d'adaptation locale.

Par ailleurs, les différentes implémentations devront être effectuées sur GPU dans le but d'atteindre des performances en temps réel. Nous comptons étudier les opérateurs de Tone mapping suivants :

- Opérateur Photographique [15] : C'est un opérateur de Tone mapping global. D'autre part, nous étudierons plusieurs aspects de cet opérateur :
 - Distinction ou non des différents types de photorécepteurs.
 - Filtrage ou non de la luminance d'adaptation.
- Opérateur Local basé sur l'opérateur proposé par Ledda et al. [?]

Opérateur Photographique

Pour cet opérateur, l'unique luminance d'adaptation est calculée de la manière suivante :

$$L_a = \exp\left(\frac{1}{N \times M} \sum_{x=1}^M \sum_{y=1}^N \log(L_w(x, y) + \epsilon)\right) \quad (13)$$

avec $L_w(x, y)$ La luminance de l'image HDR au pixel (x, y) . On utilise ϵ pour éviter des problèmes avec l'opérateur \log pour des luminances égales à zéro.

Ensuite l'opérateur applique les fonctions suivantes :

$$\begin{aligned} L_m(x, y) &= \frac{a}{L_a} L_w(x, y) \\ L_d(x, y) &= \frac{L_m(x, y)}{1 + L_m(x, y)} \end{aligned} \quad (14)$$

avec a un paramètre de l'opérateur (avec 0.18 comme valeur par défaut). Nous remarquons que l'équation 14 a une forme semblable à celle de l'équation de Naka-Rushton qui correspond à une sigmoïde.

D'autre part, nous pouvons, avec cet opérateur, faire la distinction entre les différents types de photorécepteurs. En effet, il est important de faire cette distinction car le comportement du système visuel humain change en fonction de la luminance d'adaptation. Nous pouvons définir l'opérateur de la façon suivante :

$$\begin{aligned} L_w(x, y) &= 0.256R(x, y) + 0.67G(x, y) + 0.065B(x, y) \\ L'_w(x, y) &= -0.702R(x, y) + 1.039G(x, y) + 0.433B(x, y) \end{aligned} \quad (15)$$

$$mes(x, y) = 3 \left(\frac{\log L_a(x, y) + 2}{2.6} \right)^2 - 2 \left(\frac{\log L_a(x, y) + 2}{2.6} \right)^3 \quad (16)$$

$$L_m(x, y) = \frac{a}{L_a} (mes(x, y) \times L_w(x, y) + (1 - mes(x, y)) \times L'_w(x, y)) \quad (17)$$

$$L_d(x, y) = \frac{L_m(x, y)}{1 + L_m(x, y)} \quad (18)$$

avec $mes(x, y)$ permettant d'effectuer une transition entre la vision scotopique et la vision photopique. En effet, entre ces deux visions, il se trouve la vision mesotopique où les bâtonnets et les cônes contribuent ensemble à notre perception (figure 2).

Par ailleurs, pour prendre en compte la temporalité de l'adaptation visuelle humaine, nous proposons l'utilisation d'un filtrage exponentiel (équation 19). En effet, dans la modélisation choisie, nous n'effectuons aucune distinction entre les cônes et les bâtonnets dans la luminance d'adaptation. Il serait intéressant de regarder d'autres modélisations comme celle proposée par Pattanaik et al. [14] qui repose elle aussi sur des filtres exponentiels et qui intègre cette distinction.

$$L_a(t) = (1 - \alpha) \times L_a(t - 1) + \alpha \times L_a \quad (19)$$

avec $L_a(t)$ la luminance d'adaptation lissée avec un filtre exponentiel, L_a la luminance d'adaptation calculée grâce à l'équation 13 et α le paramètre de lissage du filtre.

Enfin, pour convertir L_d en représentation couleur, on utilise la valeur $mes(x, y)$ pour régler le niveau de couleur ajouté. En effet, nous voulons simuler le fait que les bâtonnets sont achromatiques et donc ne pas injecter de couleurs dans ce cas.

Opérateur Local

Dans cet article, les auteurs proposent un opérateur de Tone mapping qui peut se décomposer de la façon suivante :

- utilisation d'un opérateur de Tone Mapping de type Durand et al. [3] pour calculer la luminance d'adaptation localement à la fois pour les bâtonnets et les cônes.
- utilisation d'une adaptation temporelle prenant en compte le temps écoulé, la luminance de pré-adaptation et le temps d'exposition à cette luminance de pré-adaptation.

Pour ce faire, les auteurs partent de l'équation de Naka-Rushton pour exprimer le modèle directe (ou Forward model) :

$$V_{cone}(x, y) = \frac{L_w(x, y)^n}{L_w(x, y)^n + \sigma_{cone}(x, y)^n} \quad (20)$$

$$V_{rod}(x, y) = \frac{L'_w(x, y)^n}{L'_w(x, y)^n + \sigma_{rod}(x, y)^n}, \quad (21)$$

avec $L_w(x, y)$ et $L'_w(x, y)$ calculée avec les équations . Pour calculer les luminances d'adaptation $\sigma_{cone}(x, y)$ et $\sigma_{rod}(x, y)$, les auteurs proposent d'utiliser la relation Xie et al. (équation 2) et d'utiliser un filtrage bilatérale (équation 22) pour calculer la luminance d'adaptation.

$$J(s) = \frac{1}{k(s)} \sum_{p \in neigh(s)} f(p-s)g(I_p - I_s)I_p \quad (22)$$

$$k(s) = \sum_{p \in neigh(s)} f(p-s)g(I_p - I_s)$$

L'objectif de l'utilisation d'un filtre bilatérale est de calculer la luminance d'adaptation pour chaque pixel de l'image. D'autre part, il permet d'éviter l'apparition d'artefacts de type Halo. Pour ce genre de filtre, on utilise souvent une fonction gaussienne (équation 23) pour les fonctions f et g.

$$g(I_p - I_s) = \exp\left(\frac{(I_p - I_s)^2}{2 * \sigma^2}\right) \quad (23)$$

Ensuite, ils appliquent le modèle inverse (ou Reverse model) pour calculer la luminance affichable (équation 8). Notons que la vision mésopique est aussi prise en compte. Pour ce faire, les auteurs définissent un facteur mes(x,y) (équation 16) compris entre [0, 1] permettant une transition entre les visions scotopique et photopique.

$$L_{d,cone} = mes(x, y)L_{d,mean} \left(\frac{V_{cone}(x, y)}{1 - V_{cone}(x, y)} \right)^{\frac{1}{m}} \quad (24)$$

$$L_{d,rod} = (1 - mes(x, y))L_{d,mean} \left(\frac{V_{rod}(x, y)}{1 - V_{rod}(x, y)} \right)^{\frac{1}{m}}$$

Pour effectuer l'adaptation visuelle en prenant la dimension temporelle de celle-ci, les auteurs proposent d'effectuer un filtrage sur la luminance d'adaptation. Ils proposent une solution pour modéliser le comportement des cônes et des bâtonnets dans la "dark adaptation" et "light adaptation". Par exemple, pour la "dark adpation" au niveau des cônes, les auteurs l'effectuent avec l'équation 25.

$$\sigma_t(x, y) = \sigma_{fin}(x, y) + \frac{CONEdark - postdark_{fin}}{postdark_{in} - postdark_{fin}}(\sigma_{in}(x, y) - \sigma_{fin}(x, y)) \quad (25)$$

avec $postdark_{in}$ et $postdark_{fin}$ des constantes du modèle (voir article pour plus de détails), $\sigma_{in}(x, y)$ la luminance d'adaptation avant le changement d'illumination, $\sigma_{fin}(x, y)$ la luminance d'adaptation considérant que l'utilisateur est adapté à la nouvelle illumination et :

$$CONE_{dark} = \sigma_{in}(x, y)e^{K/t}$$

avec K un paramètre de leur modèle et t le temps écoulé depuis la dernière variation d'illumination.

Cependant, le problème avec cette modélisation est qu'elle ne peut s'appliquer que pour des images fixes. En effet, on peut utiliser leur modélisation en prenant en entrée deux images HDR :

- Représentant l'image ou l'utilisateur est adapté avant la variation de l'illumination (pour calculer σ_{in}).
- Représentant l'image ou il y a eu le changement d'illumination (pour calculer σ_{fin}).

Le problème est que notre opérateur de Tone mapping évoluera dans un environnement dynamique où l'utilisateur pourra interagir avec celui-ci. Donc il nous est impossible de produire ces deux images. Une solutions serait de transformer leur modèle pour l'adaptation temporelle des filtres exponentiels. Une solution peut être d'utiliser la modélisation que propose Pattanaik et al. [14] pour arriver à construire un nouveau modèle d'opérateur de Tone mapping. Cette piste devra être développée dans la suite du stage.

4.3 Balance des blancs

Pour ce type opérateur, nous voulons étudier plusieurs techniques différentes et évaluer la qualité globale de la solution avec ces opérateurs. Ces opérateurs interviendront sur l'image LDR après les opérateurs de Tone mapping.

Grey world

Cette méthode a été proposée par Buchsbaum [1]. Elle se base sur deux hypothèses :

- il n'y a qu'une seule source de lumière dans la scène,
- si la scène est assez complexe, alors la réflectance moyenne dans la scène est achromatique.

Si ces deux hypothèses sont respectées, l'illuminant de référence peut, dans ce cas, être estimé comme la moyenne des luminances de toute l'image. Soit $f = (R, G, B)^T$, $f(i)$ la valeur colorimétrique du pixel i et n le nombre de pixels de l'image :

$$f_w = \frac{1}{n} \sum_{i=1}^n f(i) \quad (26)$$

Par ailleurs, nous allons implémenter cette méthode sur GPU. Pour ce faire, nous utilisons le mip-mapping pour calculer f_w . En effet, en réduisant l'image, l'opération de mipmapping effectue une opération bilinéaire qui consiste à effectuer des moyennes entre 4 pixels voisins. Ceci permet de sous-échantillonner la texture en divisant par 4 sa taille pour chaque niveau de filtrage. Le dernier niveau de filtrage nous donne une texture 1x1. Ensuite, grâce à un shader, on récupère f_w et on effectue la correction colorimétrique en utilisant la transformée de Bradford. [7].

Méthode Cozot

La méthode proposée par les auteurs se base sur l'hypothèse de Von Kries : l'adaptation chromatique ou balance des blancs s'effectue au niveau de l'œil en fonction de l'éclairement arrivant à l'œil. L'éclairement est calculé en prenant en compte l'éclairage direct de l'œil et l'éclairage indirect provenant des objets dans le champ de vision. La couleur de l'éclairement arrivant à l'œil est alors utilisée comme couleur du blanc de référence. Cette méthode a les caractéristiques suivantes :

- prise en compte de l'éclairage direct,
- prise en compte de l'éclairage indirect et d'éventuels milieux semi transparents (comme une vitre teintée).

Donc, les auteurs proposent d'utiliser une demi-sphère au niveau de la position de l'œil pour capturer l'illumination incidente entrante dans l'œil. Pour ce faire, nous effectuons le rendu de cette demi-sphère comme l'on effectuerait le rendu pour calculer l'éclairement pour les objets de la scène. Nous stockons le résultat dans un buffer, puis nous utilisons le mip-mapping pour calculer l'illuminant de référence.

Méthode Wilkie

La technique proposée par Wilkie et al. [19] est relativement simple. Elle est résumée dans la figure 20.

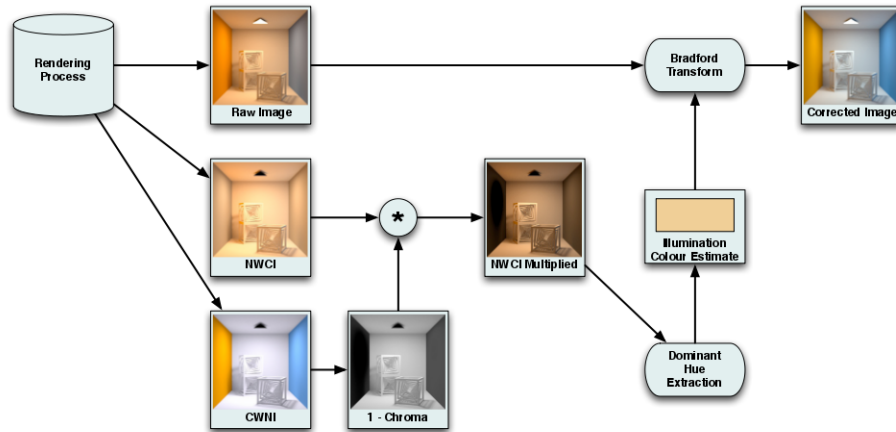


FIGURE 20 – L’algorithme total

Cette technique repose sur deux images (CWNI et NWCI) pour déterminer l’illuminant de référence et ensuite corriger le rendu de la scène à l’aide d’un algorithme d’illumination globale. Les deux images sont :

- CWNI (Coloured World Neutralised Illuminants) : image dans laquelle l’illumination directe et indirecte est neutre. C’est-à-dire que la luminance des sources est remplacée par leur luminance achromatique équivalente (blanche).
- NWCI (Neutralised World Coloured Illuminants) : image où toutes les BRDF seraient blanches, cela revient à imposer aux objets d’avoir une couleur intrinsèque blanche.

Après avoir calculé les deux images, la seconde étape consiste à extraire l’illuminant de référence. Pour ce faire, on effectue un traitement de l’image CWNI pour extraire tous les matériaux blancs qui reçoivent un éclairage. Pour ce traitement, les auteurs utilisent une transformation appelée "1 - Chroma" (voir papier pour plus de détails).

Pour finir, les auteurs composent l’image NWCI représentant la couleur de l’éclairage avec l’image CWNI ayant subi la transformée "1 - Chroma". Le résultat de cette composition ("NWCI Multiplied") peut être défini comme l’image où seuls les objets blancs reçoivent l’éclairage. Enfin, pour extraire l’illuminant de référence, les auteurs effectuent une moyenne sur toute l’image "NWCI Multiplied".

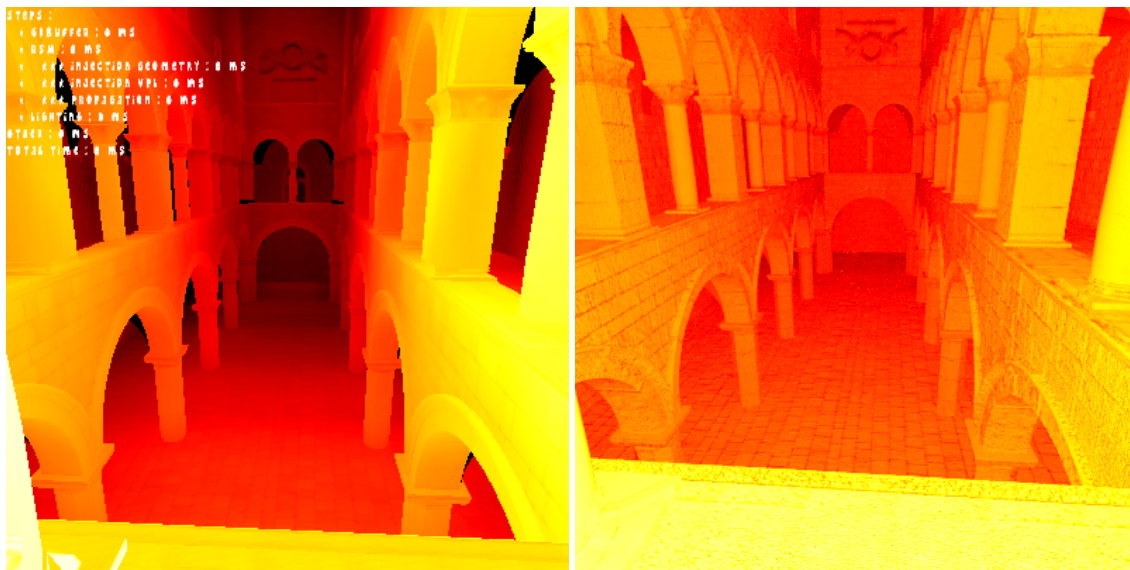
Donc, pour implémenter cette technique, nous proposons d’effectuer deux calculs séparés dans l’illumination globale. En effet, les auteurs proposent de modifier les techniques d’illumination globale pour éviter un coût trop important de leur technique. Cependant, cette modification est très difficile dans le cadre des LPV à cause de l’utilisation des harmoniques sphériques et de la phase de propagation.

5 Résultats

5.1 Extension rendu HDR

Pour vérifier les résultats produits par le moteur d'illumination globale, nous avons décidé de faire une validation en utilisant un autre moteur d'illumination sur la même scène. Nous avons choisi d'utiliser PBRT¹² développé par Matt Pharr et Greg Humphrey pour effectuer cette validation.

Dans ce but, nous avons utilisé la version de la "sponza scene" simplifiée car elle était disponible pour les différents moteurs de rendu. De plus, nous avons essayé d'obtenir les paramètres les plus proches entre les deux moteurs. Enfin, pour visualiser les luminances entre 1 et 10^5 , nous avons utilisé le même rendu fausse couleur que pour l'image 6.



(a) Résultat LPV (temps de calcul 1/30 s)

(b) Résultat PBRT (temps de calcul 1920 s)

FIGURE 21 – Les deux rendus pour la scène sponza. Certaines zones noires sur le rendu LPV sont dues au modèle.

On remarque que les rendus ne sont pas identiques. Ceci est dû au nombre d'approximations qu'effectue la méthode de CryTek. Cependant, on retrouve la structure dans le rendu avec un fond recevant moins d'énergie que les objets en premiers plan.

Toutefois, pour effectuer une validation correcte, il faudrait implémenter un chargeur de scènes PBRT. En effet, même si la géométrie des deux scènes est quasiment la même, il reste des différences surtout au niveau des matériaux.

12. Moteur de rendu disponible à l'adresse suivante : <http://www.pbrt.org/>

5.2 Opérateur de Tone mapping

Nous avons rencontré un problème lors de l'implémentation des opérateurs de Tone mapping. En effet, nous avons décidé d'implémenter d'abord l'opérateur photographique et d'utiliser le mipmapping pour effectuer le calcul de la luminance d'adaptation. Cependant, nous nous sommes rendus compte que l'opération de mipmapping n'était pas disponible sur notre carte graphique pour des textures flottantes (GL_RGBA32F).

Une solution à ce problème serait de passer par les shaders pour effectuer la moyenne. En effet, nous pourrions utiliser plusieurs passes pour effectuer des moyennes successives en réduisant la taille de la texture (figure 22).

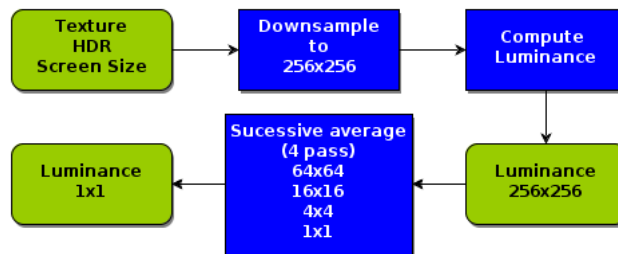


FIGURE 22 – Mipmapping à base de shaders. On utilise un sous-échantillonnage à 256x256 pour accélérer les calculs.

Donc, dans la suite du stage, il nous restera à implémenter cette solution et à implémenter les différents modèles d'opérateurs de Tone mapping proposés.

5.3 Balance des blancs

Pour les opérateurs de balance des blancs, nous avons, jusqu'à maintenant, implémenté uniquement le Grey world sur CPU et GPU. Cependant, nous avons observé des différences de résultats entre les deux approches (figure 23). Nous pensons que le problème vient du fait de l'utilisation du mipmapping pour calculer la moyenne de l'image. Une solution à ce problème serait d'implémenter un mipmapping par les shaders (figure 22).



(a) Résultat sans balance des blancs (b) Résultat après l'application de la méthode Grey world sous CPU (c) Résultat après l'application de la méthode Grey world sous GPU

FIGURE 23 – Montre les différents résultats obtenu avec l'approche CPU et l'approche GPU

6 Perspectives et Bilan

Il reste encore beaucoup de travail à accomplir. Cependant, étant donné que je suis à la moitié de mon stage, il va m'être encore possible d'explorer de nombreuses pistes. Voici un tableau récapitulatif de l'avancement des travaux :

	Étude	Modélisation	Implémentation	Validation
Prérequis implémentation	x	x	x	x
LPV	x	x	x	(x) ¹³
Opérateur de Tone Mapping				
- Photographic Tone Mapping	x	x		
- Local Tone Mapping	x			
Opérateur de balance des blancs				
- Grey world	x	x	x	
- Méthode Cozot	x	x		
- Méthode de wilkie	x	x		

6.1 Perspectives

6.1.1 Travaux sur le moteur expérimental

Je pense qu'il est important d'avoir de bons outils pour effectuer des travaux expérimentaux. C'est pour cette raison que je continuerai les travaux effectués sur le moteur de rendu expérimental. Les performances de celui-ci ne sont pas encore optimales et il reste du travail à effectuer. Les différents axes des travail seront les suivants :

- Amélioration du graphe de scène et implémentation de différents algorithmes de culling dans le but d'améliorer les performances globales du moteurs de rendu.
- Possibilité de spécifier des scènes plus finement comme PBRT propose de le faire.
- Implémenter une autre méthode d'illumination globale. Il serait, par exemple, intéressant d'implémenter une méthode d'illumination globale basée sur des données pré-calculées pour fournir des résultats plus précis.

6.1.2 Étude d'autres modèles Tone mapping

Parmi les opérateurs proposés pendant ce stage, nous utilisons uniquement des opérateurs ne prenant pas en compte la spécificité du comportement du système visuel humain pour des scènes mesotopiques. En effet, dans notre solution, nous utilisons simplement une combinaison linéaire pour modéliser le comportement des opérateurs de Tone mapping. Ce type de scène est difficile à modéliser car, dans ce cas de figure, les cônes et les bâtonnets ont une sensibilité. Plusieurs projets comme le projet MOVE¹⁴ ou la méthode proposée par Krik et al [8] tentent de modéliser l'opérateur de Tone mapping dans ces conditions.

13. En relation avec ce qui a été dit dans la partie "Extension rendu HDR", la validation doit être plus poussée.

14. MOVE : Mesotopic Optimisation of Visual Efficiency. Rapport relatif à l'étude des conditions mesotopiques dans le cadre de la conduite : http://www.lightinglab.fi/CIETC1-58/files/MOVE_Report.pdf

D'autre part, il serait aussi intéressant d'étudier les opérateurs de Tone mapping non basés sur des modèles biologiques. En particulier les modèles basés sur les réponses des films de caméra comme cela se fait dans le jeu Star Ocean 4¹⁵. Ceci nous paraît important car notre objectif est de produire un résultat réaliste et il se peut que le fait que l'utilisateur regarde le résultat à travers un système de restitution type écran le conduise à se mettre à la place d'une caméra virtuelle.

6.1.3 Intégration dans un opérateur commun

Pour l'instant, avec la solution proposée, nous produisons l'image finale en respectant les étapes suivantes :

- On produit une image HDR avec notre moteur d'illumination globale.
- On transforme l'image HDR en image LDR en utilisant un opérateur de Tone mapping.
- On corrige la couleur de l'image LDR avec un opérateur de balance des blancs.

Par conséquent, dans notre cas, les deux dernières étapes qui ont pour but de modéliser le système visuel humain sont effectuées séquentiellement. Cependant, il est pratiquement certain que notre système visuel humain effectue toutes ces étapes en un seul processus. Par exemple, le rôle de l'illuminant de référence peut jouer un rôle dans l'opération de Tone mapping car celui-ci pourrait impacter la façon de calculer la luminance. Donc, il faudrait étudier la possibilité d'ajouter, dans l'étape de Tone mapping, l'illuminant de référence que l'on utiliserait aussi pour la balance des blancs.

Enfin, il serait intéressant d'étudier plus en détail des modèles comme iCAM06 [9] qui proposent de prendre en compte un nombre plus important de spécificités du système visuel humain.

6.2 Bilan

Pour effectuer ce stage, nous avons dû mettre en place un moteur d'illumination globale. Ce travail a été long du fait de la complexité de la tâche. De plus, certains points dans la modélisation sont encore à l'étude pour que la solution proposée fonctionne dans l'environnement interactif.

D'un point de vue personnel, ce stage m'a beaucoup apporté. En effet, ce stage m'a permis de mieux comprendre comment est organisée la recherche au États Unis. En effet, celle-ci n'est pas organisée de la même manière qu'en France. Par exemple, les chercheurs sont très peu regroupés en équipe de recherche et prônent plutôt un travail individuel. Plus globalement, ce stage m'a permis de me confronter avec une autre culture et d'améliorer mes compétences de communication en anglais.

En conclusion, ce stage est riche d'enseignement car beaucoup d'autres problèmes ont surgi, ce qui nous a conduit à reproblématiser nos questions de recherche. Ainsi, l'objectif pour la fin du stage est de proposer une version complète de notre opérateur (Tone mapping et balance des blancs) avec une phase de validation. Pour la validation, une possibilité serait d'utiliser un dispositif de restitution d'images HDR et de le comparer avec notre solution.

15. Présentation de leur solution de Tone mapping disponible à l'adresse suivante : http://research.tri-ace.com/Data/SO4_flexible_shader_management_and_postprocessing.ppt slides 88.

Références

- [1] G.. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310, 1980.
- [2] Carsten Dachsbacher and Marc Stamminger. Reflective shadow maps. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games, I3D '05*, pages 203–231, New York, NY, USA, 2005. ACM.
- [3] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph.*, 21 :257–266, July 2002.
- [4] Sumanta Pattanaik Paul Debec Wolfgang Heidrich Erik Reinhard, Greg Ward and Karol Myszkowski. *High Dynamic Range Imaging, Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2010.
- [5] Anton Kaplanyan. Light propagation volumes in cryengine 3, 2009.
- [6] Anton Kaplanyan and Carsten Dachsbacher. Cascaded light propagation volumes for real-time indirect illumination. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games, I3D '10*, pages 99–107, New York, NY, USA, 2010. ACM.
- [7] Naoya Katoh and Kiyotaka Nakabayashi. Applying mixed adaptation to various chromatic adaptation transforms. pages 299–305, 2001.
- [8] Adam G. Kirk, , and James F. O'Brien. Perceptually based tone mapping for low-light conditions. *ACM Transactions on Graphics*, 30(4) :42 :1–10, July 2011. Proceedings of ACM SIGGRAPH 2011, Vancouver, BC Canada.
- [9] Jiangtao Kuang, Garrett M. Johnson, and Mark D. Fairchild. icam06 : A refined image appearance model for hdr image rendering. *J. Vis. Comun. Image Represent.*, 18 :406–414, October 2007.
- [10] Patrick Ledda and et al. A local model of eye adaptation for high dynamic range images. In *proceedings of ACM Afrigraph '04*, pages 151–160. ACM Press, 2004.
- [11] Martin Mittring. Finding next gen : Cryengine 2. In *ACM SIGGRAPH 2007 courses, SIGGRAPH '07*, pages 97–121, New York, NY, USA, 2007. ACM.
- [12] Y. N. Morone, M. D. Fairchild, R. W. G. Hunt, C. Li, M. R. Lou, and T. Newman. The ciecam02 color appearance model. In *In Color Imaging Conference (2002), IS&T*, pages 23–27. Society for Imaging Science and Technology, 2002.
- [13] Laszlo Neumann, Francesc Castro, Attila Neumann, and Mateu Sbert. Color appearance in multispectral radiosity. In G. Renner L. Szirmay-Kalos, editor, *Proceedings on the 2nd Hungarian Computergraphics and Geometry Conference*, pages 183–194, 2003.
- [14] Sumanta N. Pattanaik, Jack Tumblin, Hector Yee, and Donald P. Greenberg. Time-dependent visual adaptation for fast realistic image display. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, pages 47–54, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

- [15] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 267–276, New York, NY, USA, 2002. ACM.
- [16] Kadi Bouatouch Rémi Cozot and Sébastien Hillaire. Color adaptation in global illumination, En cours.
- [17] Sinje Thiedemann, Niklas Henrich, Thorsten Grosch, and Stefan Muller. Voxel-based global illumination. In *Symposium on Interactive 3D Graphics and Games*, I3D '11, pages 103–110, New York, NY, USA, 2011. ACM.
- [18] J. Von Kries. *Chromatic adaptation*, pages 109–119. MIT Press, [1905] 1970.
- [19] A. Wilkie and A. Weidlich. A robust illumination estimate for chromatic adaptation in rendered images. In *Eurographics Symposium on Rendering 2009*, 2009.
- [20] Z. Xie and T.G. Stockham. Toward the unification of three visual laws and two visual models in brightness perception. *Systems, Man and Cybernetics, IEEE Transactions*, 19, 1989.