



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre :IVA10/IVA/M2/2023

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Image et vie artificielle**

Modélisation d'une scène 3D avec des ombres douces et des réflexions à l'aide de la technique Raymarching basée sur GPU

Par :

BEN SAHLA MOHAMED RAMI

Soutenu le 20/06/2023 devant le jury composé de :

Fakraoui Farah

MCB

Président

Zerari Abdelmoumen

MCB

Rapporteur

Boucetta Mebarek

MAA

Examineur

Année universitaire 2022-2023

Remerciements

Mon premier remerciement va a Allah de m'avoir donné toute cette force.

Je tenais à remercier vivement mon encadreur, Monsieur ABD EL MOUMÉNE ZERARI, pour la confiance qu'il m'a accordée, pour son aide précieuse et pour le temps qu'il a bien voulu me consacrer, ainsi que pour la liberté concédée quant a mes orientations de recherche.

Je remercie également les membres du jury : Fakraoui Farah et Boucetta Mebarek d'avoir accepte l'évaluation de ce travail.

Je remercie ma famille pour leur grand soutien et mes meilleurs amis pour m'avoir motivé cette année.

Je voudrais aussi remercier tous les enseignants du département d'informatique, qui m'ont forme durant la période d'étude à l'université Mohamed Khider – Biskra.

Mes sincères remerciements s'adressent à mes collègues de département informatique Biskra.

Résumé

Dans ce mémoire, nous nous intéressons à la modélisation d'une scène 3D en mettant l'accent sur les reflets et les ombres douces en présence d'éclairage.

Ce procédé est appliqué par la technique Ray marching au GPU et cette technique est une méthode de rendu qui utilise des rayons pour calculer l'interaction de la lumière avec les objets d'une scène 3D, L'utilisation du GPU pour accélérer le rendu en temps réel.

Le travail essentiel de notre application consiste à modéliser un modèle 3D par la géométrie implicite, puis appliqué le lancer de rayon et le SDF, après on calcul (BRDF) l'éclairage par le modèle de Blinn-Phong.

Notre objectif est d'obtenir un scène 3D par le GPU, on fait une accélération de temps de calcul sans sacrifier la qualité de l'image, le résultat sera projeter sur l'écran pour obtenir l'image finale.

Mots clés : Ray marching,reflets,ombres douces,GPU,BRDF,Blinn-Phong.

Abstract

In this thesis, we are interested in modeling a 3D scene with an emphasis on reflections and soft shadows in the presence of lighting.

This process is applied by the Ray marching technique to the GPU and this technique is a rendering method that uses rays to calculate the interaction of light with objects in a 3D scene, Using the GPU to speed up rendering in time real.

The essential work of our application consists in modeling a 3D model by the implicit geometry, then applying ray tracing and the SDF, after calculating (BRDF) the lighting by the Blinn-Phong model.

Our objective is to obtain a 3D scene by the GPU, we accelerate the calculation time without sacrificing the quality of the image, the result will be projected on the screen to obtain the final image.

Keywords : Ray marching, reflections, soft shadows, GPU, BRDF, Blinn-Phong.

Table des matières

I	Illumination et modélisation	12
I.1	Introduction	13
I.2	Illumination	13
I.2.1	Illumination locale	13
I.2.1.1	Modèle d'éclairage de phong	14
I.2.2	Illumination globale	15
I.2.2.1	Radiosité	15
I.2.2.2	Path tracing	16
I.3	L'ombrage	17
I.4	Ombre	17
I.4.1	Ombre dure	17
I.4.2	Ombre Douce	18
I.4.3	La relation entre la source et l'ombre	18
I.4.4	Problèmes importants dans le calcul des ombres douces	19
I.5	Réflexion	20
I.5.1	Environnement mapping	20
I.5.2	Technique de l'environnement mapping	20
I.5.2.1	Environnement mapping sphérique	20
I.5.2.2	Environnement mapping cubique	21
I.5.2.3	Environnement mapping double parabolöide	22
I.5.3	Réflexion de l'environnement mapping	23
I.6	La Modélisation	25
I.6.1	Base de points	25
I.6.2	Base de images	26
I.6.3	Base de surface implicite	26
I.7	Conclusion	27

II	Ray Marching	28
II.1	Introduction	29
II.2	Concept de base	29
II.3	Conception de Ray Marching	30
II.4	Version CPU	31
II.5	Version GPU	32
II.6	La Différence entre la version CPU et la version GPU	33
II.7	Les travaux connexe	33
II.7.1	Ray Marching and Signed Distance Functions	33
II.7.2	Ray Marching de Michael walczyk	34
II.7.3	Une boîte à outils d'unité hautement interactive pour la construction visuelle de champs de distance signés	37
II.7.4	Ray Marching de Tyler Bovenzi	38
II.7.5	Ray Marching Distance Fields with CUDA	39
II.8	Conclusion	39
III	Mise en œuvre et résultats	41
III.1	Introduction	42
III.2	Environnements et outils	42
III.2.1	la première partie	42
III.2.1.1	Matériel de configuration	42
III.2.2	la deuxième partie	43
III.2.2.1	OpenGL GLSL	43
III.3	Architecture générale	43
III.3.1	La géométrie implicite	44
III.3.2	Lancer de rayons	44
III.3.3	La fonction de distance signée (SDF)	44
III.3.4	Algorithme de Ray Marching	45
III.3.5	BRDF	45
III.3.6	Calcul d'illumination et l'ombre	45
III.3.7	Scène finale	46
III.4	Description détaillé du système	46
III.4.1	Les Fichiers Shaders	46
III.4.2	La fonction de distance signée	46
III.4.2.1	La fonction de distance signée d'une sphère	46
III.4.2.2	La fonction de distance signée d'une plan	47
III.4.2.3	La fonction de distance signée (SDF) d'une sphère et d'un plan	48

III.4.2.4	La fonction estimateNormal	49
III.4.3	La Fonction de distribution bidirectionnelle de la réflectance	50
III.4.3.1	La modèle d'éclairage Blinn-Phong	50
III.4.3.2	L'illumination d'un point donné sur une surface	51
III.4.3.3	La direction d'un rayon	51
III.4.4	Les calculés d'Algorithme de Ray Marching	52
III.4.5	Les paramètres de couleur damier	53
III.5	Les résultats	53
III.5.1	La scène finale	54
III.5.2	Des résultats sans l'apparition de damier sur le sphère	55
III.5.3	Des résultats avec l'apparition de damier sur le sphère	56
III.5.4	Des résultats avec l'images par seconde	56
III.6	Analyse des résultats	57
III.7	Perspectives	57

Table des figures

I.1	Illumination locale [4]	14
I.2	Illumination globale [14]	15
I.3	Modélisation d'illumination globale avec la radiosité [1]	16
I.4	Ombre Dure [23]	18
I.5	Ombre Douce [23]	18
I.6	Sphère Mapping [31]	21
I.7	Cube Map Texture [34]	22
I.8	Map Double Paraboloid [18]	23
I.9	Exemple de Réflexion Mapping [37]	24
I.10	Réflexion mapping [37]	25
I.11	Modélisation par les surfaces implicites [29]	27
II.1	Lancer de rayons [21]	30
II.2	La Version CPU de Ray Marching [32]	32
II.3	La Version GPU de Ray Marching [16]	33
II.4	Conception de Jamie wong [38]	34
II.5	La Conception de Michael Walczyk [35]	35
II.6	Circle-Distance [35]	35
II.7	La Génération de Michael Walczyk [35]	36
II.8	Déformer la fonction de distance [35]	37
II.9	Boîte à outils de Raymarching pour Unity [36]	38
II.10	Ombrage des Distances [9]	39
III.1	L'architecture Générale du Système	43
III.2	Diagramme de La fonction de distance signée d'une sphère	47
III.3	Diagramme de La fonction de distance signée d'un plane	48
III.4	Diagramme de La fonction de distance signée d'un plane et une sphère	49
III.5	Diagramme de contribution à la lumière	50
III.6	Diagramme de La Fonction d'Éclairage Phong-Blinn	51

III.7 Diagramme de L'Algorithme de Ray Marching	52
III.8 Diagramme de damier	53
III.9 Le résultat final de shader	54
III.10Des résultats sans l'apparition de damier sur le sphère	55
III.11Des résultats avec l'apparition de damier sur le sphère	56
III.12fps	57
III.13Travaux futurs à travers lesquels l'amélioration et le développement peuvent être réalisés . . .	58
article graphicx	

Introduction générale

La modélisation de scènes 3D réalistes est un défi complexe en infographie. Les ombres douces et les reflets sont des ingrédients essentiels pour créer des rendus visuellement agréables. Les techniques de marche de rayons basées sur GPU se sont révélées être une approche prometteuse pour obtenir ces effets de manière efficace et réaliste.

Ray marching est une méthode de rendu qui utilise des rayons pour traverser une scène 3D et calculer comment la lumière interagit avec les objets. Contrairement au traçage de rayons traditionnel, qui utilise des intersections exactes avec des surfaces, la marche des rayons itère le long des rayons en évaluant les distances signées. Cette approche permet de rendre des objets complexes et de gérer des effets tels que des ombres douces et des reflets.

L'utilisation de GPU pour les techniques de marche des rayons présente de nombreux avantages. Les GPU modernes sont conçus pour effectuer des calculs massivement parallèles, de sorte que cette puissance de calcul peut être utilisée efficacement pour le rendu en temps réel.

Le problème qui se pose actuellement est Comment éviter la modélisation polygonale ?

Comment pouvons-nous utiliser le GPU afin d'accélérer le temps de calcul sans sacrifier la qualité de l'image ?

Dans ce mémoire nous nous intéressons de développer un système de modélisation de scènes 3D avec des ombres douces et des réflexions en utilisant la technique de raymarching basée sur le GPU.

Le présent mémoire est décomposé en trois chapitres :

Le premier chapitre :

Est consacré à l'étude théorique qui contient la description des modèles d'illumination, puis la définition de l'ombrage, ensuite la description des variétés d'ombre et mise au point sur les ombres douces, ainsi qu'à un aperçu sur les réflexion et l'environnement mapping. Le chapitre sera achevé par la modélisation en basée sur les surfaces implicites.

Le deuxième chapitre :

Nous commençons par donner une introduction aux raymarching, puis expliquer les concept de base, ensuite la description de la version cpu et gpu de raymarching et la différence entre les deux version, ainsi rappel de travaux similaires.

Le troisième chapitre :

Représente l'objectif et description du système, puis l'architecture générale du système et l'implémentation de notre application ainsi que les résultats obtenus. Nous terminerons à la fin par une conclusion générale et des perspectives.

Chapitre I

Illumination et modélisation

Illumination et modélisation

I.1 Introduction

L'illumination et la modélisation sont deux concepts étroitement liés dans le domaine de la création graphique et de la visualisation 3D. Ils jouent un rôle crucial dans la création d'images réalistes.

L'illumination se réfère à la façon dont la lumière interagit avec les objets dans une scène virtuelle ou réelle. Une illumination bien réalisée permet de simuler les effets d'ombre et de réflexion, donnant ainsi vie à une scène et créant une atmosphère réaliste.

La modélisation, quant à elle, se réfère à la création virtuelle d'objets et de scènes en utilisant des outils informatiques. La modélisation peut être réalisée à l'aide de logiciels spécialisés tels que des logiciels de modélisation 3D ou des moteurs de jeux vidéo. La modélisation peut également impliquer l'ajout de textures, de couleurs et de détails aux objets pour les rendre plus réalistes.

En combinant l'illumination et la modélisation, on peut créer des scènes virtuelles qui semblent réelles, avec des objets qui interagissent de manière réaliste avec la lumière. Cela permet de produire des rendus visuels de haute qualité, utilisés dans divers domaines.

I.2 Illumination

L'illumination [6] est un aspect clé du rendu en temps réel. Il s'agit du processus qui permet de simuler les effets de la lumière sur les objets dans une scène 3D. L'illumination réaliste est importante pour créer des images qui semblent naturelles et réalistes.

I.2.1 Illumination locale

L'illumination locale est une technique couramment utilisée en infographie pour simuler les effets de la lumière sur les objets dans une scène en temps réel. Cette technique prend en compte les sources de lumière à proximité des objets, telles que les lumières ponctuelles et les lumières de surface, pour créer des ombres et des reflets sur les surfaces des objets.

L'illumination locale est une technique efficace pour créer des effets de lumière et d'ombre réalistes en temps réel. Elle est couramment utilisée dans les jeux vidéo, les simulations et les applications de réalité augmentée pour créer des environnements immersifs et réalistes.

Cependant, l'illumination locale présente certaines limites. Elle ne prend pas en compte les effets de la lumière indirecte ou réfléchi, ce qui peut parfois donner des résultats peu réalistes. Pour surmonter cette limitation, des techniques plus avancées telles que l'illumination globale ont été développées pour simuler plus efficacement les effets de la lumière dans une scène.

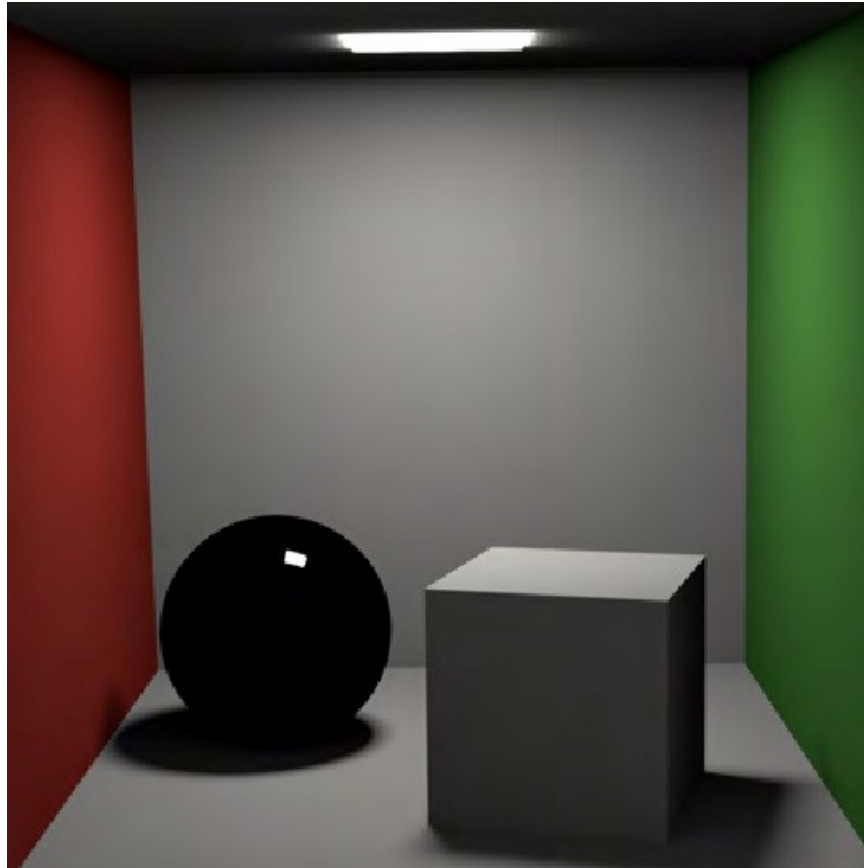


FIGURE I.1 – Illumination locale [4]

I.2.1.1 Modèle d'éclairage de phong

Le modèle d'éclairage de phong [19] est un modèle relativement simple mais très efficace pour simuler l'illumination de surfaces dans les environnements 3D en temps réel. Cependant, il présente des limites dans la représentation de la réflexion de la lumière sur les surfaces rugueuses ou translucides, pour lesquelles d'autres modèles d'éclairage peuvent être plus adaptés.

I.2.2 Illumination globale

Illumination globale est un terme utilisé dans le domaine de l'éclairage pour décrire la distribution uniforme de la lumière dans un espace donné. Il s'agit d'un concept important dans la conception d'un système d'éclairage pour assurer une visibilité et un confort visuel adéquats dans une pièce ou un espace de travail.

Pour obtenir une illumination globale [20], il est nécessaire de choisir des luminaires adaptés à l'espace en question et de les disposer de manière stratégique pour que la lumière soit répartie de manière uniforme. Les luminaires doivent être choisis en fonction des caractéristiques de l'espace, telles que la hauteur du plafond, la taille de la pièce et l'utilisation prévue.

L'illumination globale peut être mesurée en utilisant des instruments tels que des luxmètres, qui mesurent l'éclairement (la quantité de lumière par unité de surface) dans une zone donnée. Un niveau d'éclairement suffisant est essentiel pour un éclairage adéquat et confortable, et peut varier en fonction de l'activité et de la fonction de la zone éclairée.

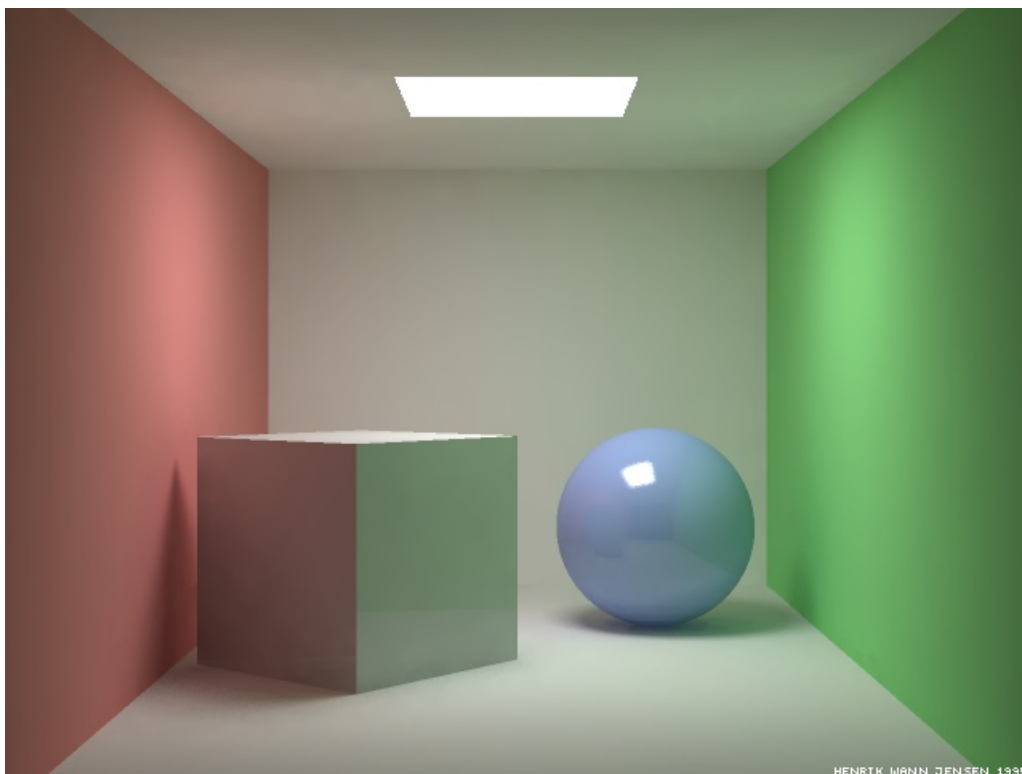


FIGURE I.2 – Illumination globale [14]

I.2.2.1 Radiosité

La radiosité calcule la quantité de lumière qui est absorbée et réfléchiée par chaque surface de la scène en utilisant une méthode de résolution de l'équation de transfert radiatif. Cette équation décrit comment la lumière se propage dans l'environnement, en prenant en compte les interactions entre les surfaces éclairées. La radiosité [27] calcule ensuite la luminance de chaque point de la scène en utilisant la quantité de lumière

réfléchi par toutes les surfaces environnantes.

La résolution du problème d'illumination globale est souvent simplifiée en utilisant la radiosité, qui représente l'intégration de la radiance. Cependant, cette méthode peut réduire le modèle d'éclairage au modèle diffus idéal, négligeant ainsi les comportements complexes des BRDFs. Bien que la radiosité soit une solution intéressante pour obtenir un rendu global convaincant, elle peut être inefficace pour représenter des détails à petite échelle.

De plus, le modèle diffus idéal utilisé avec la radiosité ne prend pas en compte les effets spéculaires tels que la réflexion et la réfraction.

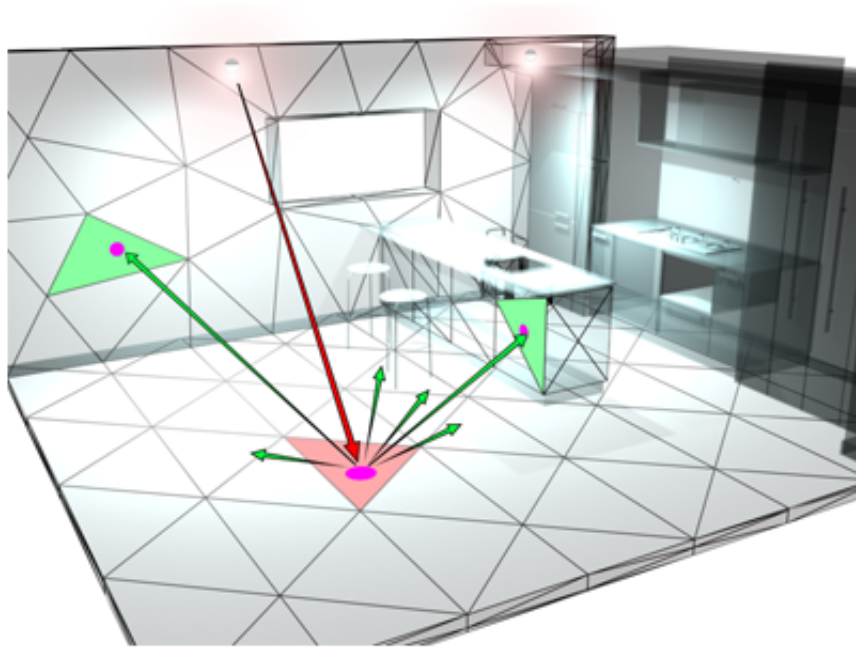


FIGURE I.3 – Modélisation d'illumination globale avec la radiosité [1]

I.2.2.2 Path tracing

Le ray tracing est une technique de rendu 3D utilisée pour calculer l'illumination globale d'une scène. Cette technique implique de lancer des rayons dans différentes directions à partir de la caméra, jusqu'à ce qu'ils atteignent une surface géométrique.

À chaque rebond, les rayons perdent de l'énergie en interagissant avec les matériaux qu'ils traversent, jusqu'à ce qu'ils atteignent une source lumineuse ou la couleur noire. Les images générées à l'aide du ray tracing sont plus photos réalistes que celles générées avec d'autres techniques, mais le calcul est plus lent et plus coûteux en termes de temps de calcul.

I.3 L'ombrage

L'ombrage [2] est une technique utilisée pour simuler des effets de lumière et d'ombre dans les images 3D en attribuant des valeurs de couleur différentes aux différentes parties d'un objet en fonction de leur position par rapport à une source lumineuse virtuelle.

Exemple : L'ombrage réaliste des personnages dans jeu vidéo.

I.4 Ombre

Une ombre est une zone sombre qui est créée lorsqu'un objet bloque la lumière ou toute autre forme de rayonnement électromagnétique. L'ombre est la zone de l'espace où la lumière ne peut pas atteindre en raison de l'obstruction créée par l'objet. Plus précisément, une ombre est créée lorsque la lumière se déplace dans une direction rectiligne et qu'un objet se trouve sur son chemin. L'objet bloque alors la lumière et crée une zone d'obscurité derrière lui, qui est l'ombre.

L'ombre [3] est généralement plus foncée près de l'objet et s'estompe graduellement à mesure qu'elle s'éloigne de l'objet. L'ombre peut être utilisée pour créer des effets artistiques dans les arts visuels, tels que la peinture ou la photographie, et est également utilisée en physique pour étudier la propagation de la lumière et des autres formes de rayonnement électromagnétique.

La présence d'une ombre indique la présence d'une source de lumière et d'un objet opaque ou semi-transparent qui bloque la lumière.

I.4.1 Ombre dure

Une ombre dure est une ombre bien définie avec une transition nette et rapide entre la zone éclairée et la zone sombre. Elle est créée lorsque la source de lumière est ponctuelle et que l'objet créant l'ombre est opaque et solide, ce qui signifie qu'il bloque complètement la lumière.

Lorsque la source de lumière est ponctuelle, comme le soleil ou un spot lumineux, les rayons lumineux se déplacent dans une seule direction, créant ainsi une ombre avec une transition nette et précise entre la zone éclairée et la zone sombre. Cette transition est dite "dure" car elle est marquée par une ligne nette de démarcation entre l'ombre et la lumière.

Les ombres dures [23] peuvent être utilisées en art pour ajouter du contraste et de la profondeur à une scène, ou en photographie pour créer des effets dramatiques. Elles peuvent également être utilisées en architecture pour créer des effets visuels, tels que des motifs d'ombre sur les murs ou les sols, ou pour fournir une protection solaire efficace en bloquant la lumière directe du soleil.

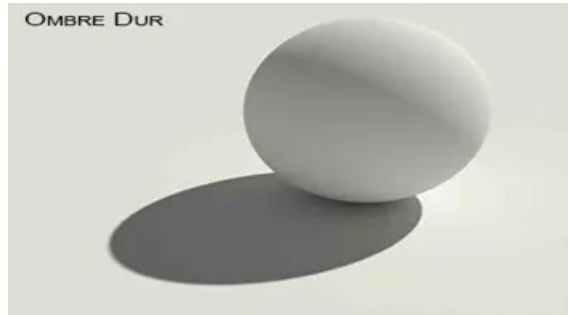


FIGURE I.4 – Ombre Dure [23]

I.4.2 Ombre Douce

Une ombre douce est une ombre qui a une transition progressive entre la zone éclairée et la zone sombre, plutôt qu'une ligne nette et précise de démarcation entre les deux zones. Elle est créée lorsque la source de lumière est étendue ou diffuse, plutôt que ponctuelle, ou lorsque l'objet créant l'ombre n'est pas complètement opaque, laissant passer une partie de la lumière.

Lorsque la source de lumière est étendue ou diffuse, les rayons lumineux sont diffusés dans plusieurs directions, créant une ombre avec une transition progressive entre la zone éclairée et la zone sombre. Cette transition est dite "douce" car elle est moins marquée et plus progressive que la transition nette d'une ombre dure.

Les ombres douces [23] peuvent être utilisées en art pour créer des effets de lumière plus subtils et plus nuancés, ou en photographie pour créer des portraits plus doux et plus flatteurs. Elles sont également utilisées en architecture pour fournir une protection solaire tout en laissant passer une certaine quantité de lumière naturelle pour éclairer l'intérieur d'un bâtiment.

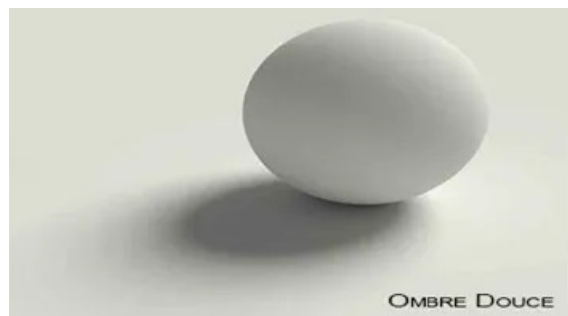


FIGURE I.5 – Ombre Douce [23]

I.4.3 La relation entre la source et l'ombre

Les ombres douces engendrées par les sources étendues [5]. On alors trois cas : - la source est totalement visible (zone éclairée) - la source est partiellement visible (zone de pénombre) - la source est complètement occultée (zone d'ombre portée).

Pour comprendre la dépendance entre la taille de la source et l'étendue de la pénombre (plus la taille de la source lumineuse est importante, plus la largeur de la zone de transition est grande).

Dans le cas plus réaliste d'une lumière surfacique, un point sur un receveur pourrait ne voir qu'une fraction de la lumière. C'est ici que se fait la distinction entre la zone complètement éclairée, la pénombre et la zone d'ombre.

Le calcul exact de ces zones est particulièrement difficile (et donc habituellement plus long), mais les ombres douces donnent des images beaucoup plus près de la réalité.

I.4.4 Problèmes importants dans le calcul des ombres douces

Le calcul des ombres douces est un problème complexe dans les domaines de la visualisation et de l'animation 3D. Voici quelques-uns des problèmes importants qui se posent lors du calcul des ombres douces :

Échantillonnage : Le calcul des ombres douces implique souvent un grand nombre de rayons qui doivent être lancés depuis la source de lumière. Cela peut être très coûteux en termes de temps de calcul. Pour réduire ce coût, les méthodes d'échantillonnage sont souvent utilisées pour calculer un sous-ensemble de rayons. Cependant, cela peut entraîner une perte de qualité dans les ombres douces.

Dépendance de la résolution : La qualité des ombres douces dépend de la résolution de l'image. Les ombres douces peuvent sembler bien sur des images haute résolution, mais deviennent floues ou pixellisées sur des images de résolution plus faible.

Difficulté à simuler les effets de la lumière indirecte : Les ombres douces sont souvent influencées par les surfaces environnantes et les matériaux réfléchissants, ce qui peut rendre leur calcul difficile. Les méthodes d'approximation sont souvent utilisées pour simuler ces effets, mais elles peuvent ne pas être suffisamment précises pour des scènes complexes.

Limitations des méthodes de rendu en temps réel : Dans les jeux vidéo et les applications de réalité virtuelle, les ombres douces doivent souvent être calculées en temps réel pour maintenir un taux de rafraîchissement fluide. Cela peut limiter la qualité et la précision des ombres douces qui peuvent être calculées. Le calcul des ombres douces est un problème complexe qui nécessite souvent des compromis entre la qualité, la précision et le temps de calcul. Les avancées récentes dans les techniques d'échantillonnage et de rendu en temps réel ont contribué à améliorer la qualité des ombres douces, mais il reste encore des défis à relever pour obtenir des résultats réalistes dans des scènes complexes.

I.5 Réflexion

I.5.1 Environnement mapping

La réflexion d'environnement [11] est une technique utilisée dans les graphiques 3D pour créer des surfaces qui semblent refléter leur environnement. Elle simule la réflexion de l'environnement sur une surface, ce qui permet de rendre l'apparence des objets plus réaliste.

Le processus d'environnement mapping consiste à capturer une image de l'environnement environnant, généralement sous forme de carte à 360 degrés, et à l'utiliser comme une texture pour la surface de l'objet. Cette texture est ensuite appliquée à la surface de l'objet, et la direction de la vue de l'observateur est utilisée pour sélectionner le pixel de la texture qui doit être appliqué à chaque point de la surface. Cela permet de simuler la réflexion de l'environnement sur la surface de l'objet.

L'environnement mapping est souvent utilisé pour rendre les surfaces réfléchissantes des objets, tels que le verre, le métal ou l'eau, plus réalistes. Il est également utilisé dans les jeux vidéo pour ajouter des effets de réflexion aux surfaces des objets, ce qui peut améliorer considérablement l'apparence visuelle du jeu.

L'environnement mapping est une technique de rendu 3D qui permet de simuler la réflexion d'un environnement à la surface d'un objet sans avoir à modéliser l'environnement en entier. Cette technique utilise une image sphérique ou cubique, appelée carte d'environnement, pour représenter l'environnement dans lequel l'objet se trouve.

I.5.2 Technique de l'environnement mapping

I.5.2.1 Environnement mapping sphérique

L'environnement mapping sphérique [30] est une technique de rendu 3D qui consiste à utiliser une image sphérique pour simuler les reflets environnants sur un objet.

Le principe de cette technique est de projeter l'image sphérique sur une sphère virtuelle qui entoure l'objet. Les rayons de lumière qui atteignent l'objet sont réfléchis selon la normale de la surface de la sphère virtuelle à l'endroit où ils la touchent. La couleur de la Texel correspondante est alors utilisée pour déterminer la couleur du pixel à rendre sur l'objet.

Cette technique est largement utilisée dans les moteurs de jeux vidéo pour simuler les réflexions environnantes sur les surfaces des objets, en particulier pour les objets métalliques ou brillants. Elle est également utilisée dans les applications de réalité virtuelle pour créer un environnement immersif et réaliste.

Cependant, cette technique présente quelques limitations, notamment pour les objets qui ne sont pas parfaitement sphériques ou pour les scènes avec des sources de lumière dynamiques. Dans ces cas, d'autres techniques de rendu peuvent être utilisées pour améliorer la qualité du rendu.



FIGURE I.6 – Sphère Mapping [31]

I.5.2.2 Environnement mapping cubique

L'environnement mapping cubique [25] est une technique de rendu 3D similaire à l'environnement MAPPING sphérique, mais qui utilise six images planes pour représenter les six faces d'un cube qui entoure l'objet.

Chaque image représente une vue de l'environnement à partir d'un point différent, correspondant à l'une des six faces du cube virtuel. Lorsque le rendu est effectué, la direction de chaque rayon de lumière qui atteint l'objet est projeté sur la face correspondante du cube virtuel, et la couleur de la texel correspondante est utilisée pour déterminer la couleur du pixel à rendre sur l'objet.

Cette technique est particulièrement utile pour les environnements qui présentent des formes anguleuses ou des bâtiments avec des façades planes, car elle permet de simuler de manière réaliste les réflexions environnantes sur ces surfaces planes. Elle est couramment utilisée dans les moteurs de jeux vidéo et les logiciels de modélisation 3D.

Cependant, cette technique peut nécessiter plus de ressources de calcul que l'environnement MAPPING sphérique en raison de la nécessité de calculer les six images planes. De plus, la qualité de l'image peut être affectée par les transitions entre les différentes faces du cube virtuel, ce qui peut causer des problèmes de discontinuité et d'aliasing.



FIGURE I.7 – Cube Map Texture [34]

I.5.2.3 Environnement mapping double parabolöide

L'environnement mapping double parabolöide est une technique de rendu 3D qui utilise deux parabolöides pour simuler les réflexions environnantes sur un objet.

Le principe de cette technique est de projeter l'image de l'environnement sur chaque parabolöide [18], en utilisant sa courbure pour déterminer la direction des rayons de lumière réfléchis. Les deux images sont ensuite combinées pour donner une représentation complète des réflexions sur l'objet.

Cette technique est particulièrement utile pour les objets qui ont une forme globale similaire à un parabolöide, comme les sphères, les cylindres ou les tasses, car elle permet de simuler les réflexions de manière très précise sur ces formes. Elle est souvent utilisée dans les jeux vidéo pour les surfaces arrondies, comme les voitures ou les personnages, mais peut également être utilisée pour les objets plus complexes.

Cependant, cette technique présente également des limitations, en particulier pour les objets qui ont des formes irrégulières ou des surfaces planes [7]. Dans ces cas, d'autres techniques de rendu peuvent être utilisées pour améliorer la qualité du rendu.

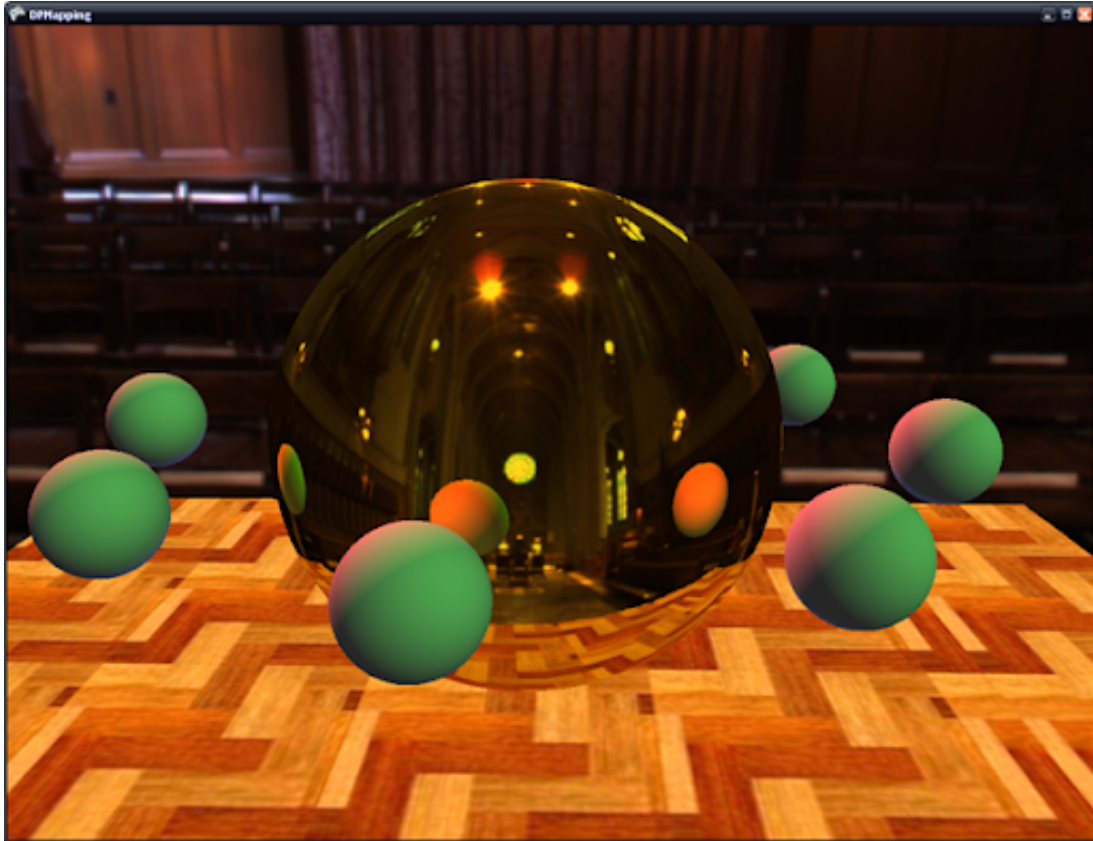


FIGURE I.8 – Map Double Paraboloïde [18]

I.5.3 Réflexion de l'environnement mapping

La réflexion environnement mapping [11] est une technique de rendu 3D qui permet de créer des effets de réflexion sur des surfaces en simulant l'environnement qui les entoure. Elle fonctionne en calculant un vecteur réfléchi par la surface, qui est utilisé pour déterminer la couleur et la luminosité de cette surface.



FIGURE I.9 – Exemple de Réflexion Mapping [37]

Pour calculer ce vecteur réfléchi, on utilise un procédé de projection qui consiste à renvoyer le vecteur d'observation de l'œil de l'observateur par rapport à un vecteur normal (perpendiculaire) à la surface. La direction miroir ainsi obtenue permet ensuite de déterminer les couleurs et les luminosités à afficher sur la surface réfléchissante.

La réflexion environnement mapping est largement utilisée dans les jeux vidéo et les films d'animation pour créer des effets de réflexion réalistes et immersifs. Elle peut être combinée avec d'autres techniques de rendu, comme la cartographie de l'occlusion ambiante et la cartographie de la lumière, pour produire des images encore plus réalistes et détaillées.

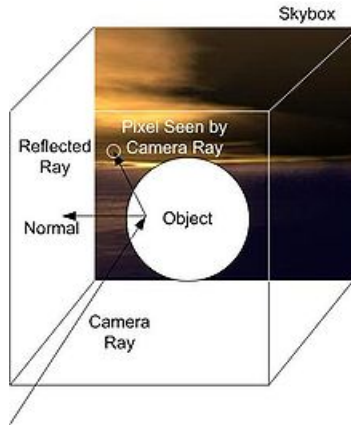


FIGURE I.10 – Réflexion mapping [37]

I.6 La Modélisation

La modélisation est le processus de création d'un modèle, qui est une représentation simplifiée d'un système réel ou d'un phénomène.

La modélisation 3D est un domaine spécifique de la modélisation qui concerne la création de représentations tridimensionnelles d'objets, de scènes ou de structures virtuelles. Elle est largement utilisée dans des domaines tels que l'animation, les jeux vidéo, l'architecture, le design de produits, la réalité virtuelle et augmentée.

La modélisation 3D demande des compétences techniques et artistiques, ainsi qu'une compréhension des principes de base de la géométrie et de la représentation spatiale.

I.6.1 Base de points

La modélisation basée sur des points [8] également connue sous le nom de modélisation de nuages de points ou modélisation de points, est une technique utilisée en infographie et en vision par ordinateur qui utilise un ensemble de points discrets dans l'espace pour représenter des objets tridimensionnels. Dans ce type de modélisation, un objet 3D est représenté par un nuage de points représentant sa surface.

La modélisation par points présente certains avantages et inconvénients par rapport aux autres techniques de modélisation 3D. En raison de la représentation discrète, un grand nombre de points peut être nécessaire pour capturer avec précision les détails de l'objet, ce qui peut entraîner une augmentation de la mémoire et de la consommation d'énergie de calcul.

Dans l'ensemble, la modélisation basée sur des points est une technique utile pour représenter des objets en trois dimensions, en particulier lorsque le niveau de détail est important.

I.6.2 Base de images

la modélisation basée sur l'image [22] est une technique puissante pour créer des modèles 3D à partir d'images 2D.

Ces informations peuvent inclure la géométrie, la structure, les textures et d'autres propriétés de l'objet.

La photogrammétrie utilise des images prises sous différents angles pour estimer la forme tridimensionnelle d'un objet.

Reconstruction à partir de plusieurs images : Avec cette approche, de nombreuses images de l'objet sont prises sous différents angles. Les images sont ensuite utilisées pour estimer la forme et la structure 3D de l'objet à l'aide de techniques d'appariement et d'enregistrement d'images. La modélisation basée sur l'image présente de nombreux avantages, notamment la facilité d'utilisation et la possibilité d'utiliser des appareils courants tels que des appareils photo numériques.

I.6.3 Base de surface implicite

La modélisation de surface implicite est une technique utilisée en infographie qui utilise des fonctions mathématiques implicites pour représenter des objets tridimensionnels.

Dans ce type de modélisation, les objets sont définis par des formules mathématiques qui décrivent la surface de l'objet. Cette équation est généralement de la forme $F(x,y,z) = 0$. où (x,y,z) sont des coordonnées spatiales et F est une fonction implicite. Ensuite, la surface de l'objet est obtenue en définissant le point (x,y,z) où $F(x,y,z)$ vaut 0. [29]

Les surfaces implicites sont excellentes pour représenter des formes complexes et organiques, car elles permettent la description de la géométrie avec des propriétés inhérentes telles qu'une courbure variable, des transitions douces entre les surfaces et des détails géométriques fins. La modélisation implicite basée sur la surface présente plusieurs avantages :

La flexibilité : Les surfaces implicites vous permettent de modéliser une grande variété de formes complexes et d'effectuer intuitivement des opérations booléennes (union, intersection, différence) entre des objets.

Représentation réaliste : Les surfaces implicites peuvent capturer des propriétés géométriques précises telles que la courbure locale, les ombres et les réflexions, permettant aux objets d'être représentés de manière réaliste.

Convivial : Puisque la surface est définie mathématiquement, nous pouvons facilement changer la forme de l'objet en ajustant les paramètres des équations implicites.

Cependant, la modélisation basée sur des surfaces implicites présente également des défis. Les inconvénients incluent :

Calcul coûteux : Les calculs pour évaluer les fonctions implicites et effectuer des opérations sur des surfaces implicites peuvent nécessiter beaucoup de calculs. Discrétisation : Rendre les surfaces implicites

utilisables dans des environnements informatiques discrets nécessite souvent de les échantillonner pour générer des maillages ou des voxels, ce qui peut entraîner une réduction de la précision et de la complexité du modèle.

La modélisation implicite [10] basée sur la surface est utilisée dans de nombreux domaines tels que la modélisation 3D, l'animation, la simulation, la réalité virtuelle et la conception assistée par ordinateur. Il offre des moyens créatifs de représenter et de manipuler des objets virtuels complexes avec des détails géométriques précis.

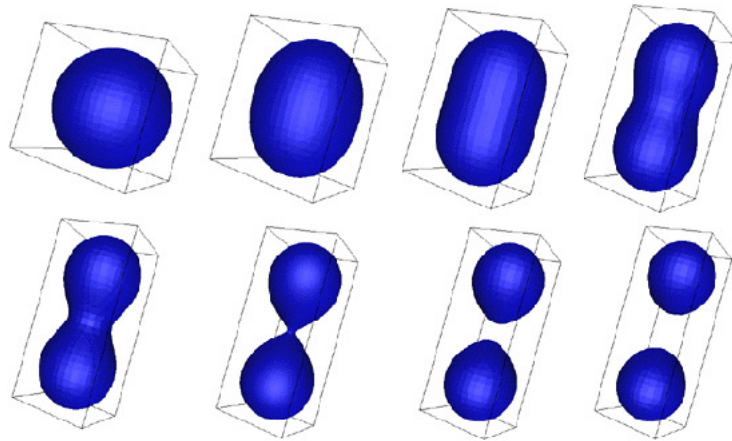


FIGURE I.11 – Modélisation par les surfaces implicites [29]

I.7 Conclusion

L'illumination et l'ombrage, ainsi que l'ombre et la réflexion environnementale sont des concepts importants en infographie et en animation 3D.

L'illumination et l'ombrage sont utilisés pour créer des effets de lumière et d'ombre sur des objets 3D afin de donner l'impression de profondeur et de réalisme. L'éclairage peut être modifié en fonction de la position de la source de lumière, de l'intensité de la lumière et de la couleur de la lumière pour créer différents effets d'ambiance.

L'ombre est un effet visuel qui se produit lorsque la lumière est bloquée par un objet, créant ainsi une zone sombre sur une surface adjacente. Les ombres peuvent être nettes ou douces en fonction de la distance entre l'objet et la surface sur laquelle l'ombre est projetée, ainsi que de la taille de l'objet.

La réflexion environnementale est utilisée pour créer des reflets réalistes sur les surfaces d'objets 3D. Elle est basée sur la façon dont les surfaces réfléchissent la lumière et peut être modifiée en fonction de la texture de la surface et de l'environnement qui l'entoure.

En conclusion, l'illumination et l'ombrage, l'ombre et la réflexion environnementale sont des techniques importantes pour créer des images 3D réalistes et convaincantes. En utilisant ces concepts, les artistes et les développeurs peuvent créer des environnements virtuels immersifs qui semblent réels et donnent vie à des objets et des personnages virtuels.

Chapitre II

Ray Marching

Ray Marching

II.1 Introduction

Ray Marching [13] est une technique utilisée en infographie pour le rendu de scènes 3D. Cela implique de tracer un rayon de la caméra à travers chaque pixel du plan image et de marcher le long du rayon par petites étapes jusqu'à ce que le rayon croise un objet dans la scène.

La marche se fait en boucle, où à chaque étape, la position du rayon est vérifiée par rapport aux objets de la scène pour déterminer s'il s'est croisé avec l'un d'entre eux. Si une intersection est trouvée, la couleur du pixel est calculée en fonction des propriétés de l'objet au point d'intersection, telles que son matériau et son éclairage.

Ray marching est souvent utilisée pour rendre des objets complexes, tels que des fractales et des données volumétriques, qui ne peuvent pas être facilement représentés à l'aide de primitives géométriques traditionnelles. La technique est particulièrement bien adaptée pour représenter ces types d'objets car elle peut gérer les formes irrégulières et les motifs auto-similaires qui sont courants dans les fractales et les données volumétriques. De plus, la marche des rayons est relativement simple à mettre en œuvre et peut être effectuée efficacement sur du matériel graphique moderne.

L'un des défis de la marche des rayons est de déterminer la taille des pas pour marcher le long du rayon. Si la taille du pas est trop grande, le rayon peut manquer de petites caractéristiques de l'objet en cours de rendu, ce qui entraîne des artefacts visuels. D'autre part, si la taille du pas est trop petite, l'algorithme peut devenir trop coûteux en calculs. Par conséquent, trouver la taille de pas optimale peut nécessiter des expérimentations et des ajustements.

II.2 Concept de base

Pour maîtriser la marche des rayons, il y a plusieurs concepts et compétences clés à apprendre :

Lancer de rayons

Comprendre les principes fondamentaux du lancer de rayons est essentiel pour la marche des rayons [15]. Cela inclut de comprendre comment projeter un rayon de la caméra à travers un pixel sur le plan de l'image et comment calculer le point d'intersection du rayon avec un objet dans la scène.

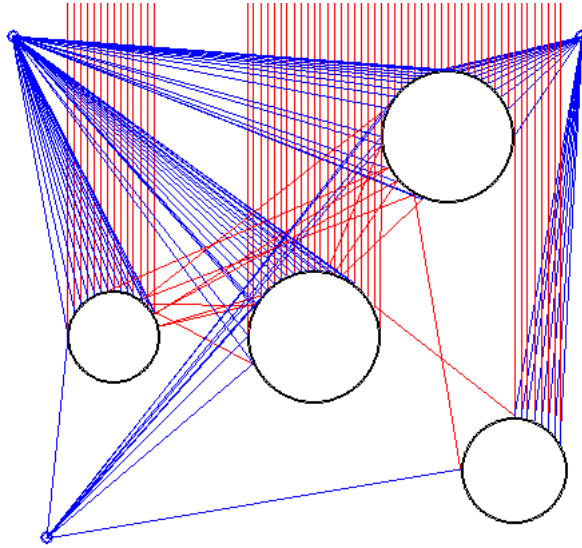


FIGURE II.1 – Lancer de rayons [21]

Fonctions de distance signées

Les fonctions de distance signées [38] sont des fonctions mathématiques qui peuvent être utilisées pour représenter des formes complexes de manière simple. Ils sont largement utilisés dans la marche des rayons, et leur maîtrise est essentielle pour créer des scènes complexes et visuellement saisissantes.

Éclairage

Il est important de comprendre les principes de l'éclairage pour créer des scènes réalistes [26] et visuellement attrayantes. Cela inclut la compréhension des différents types de modèles d'éclairage, tels que l'éclairage ambiant, diffus et spéculaire, et la manière de les implémenter dans un algorithme de marche de rayons.

Optimisation

L'optimisation des algorithmes de marche des rayons [28] est importante pour obtenir des fréquences d'images élevées et un rendu en temps réel. Cela inclut des techniques telles que la terminaison précoce et les tailles de pas adaptatives, qui peuvent aider à réduire le coût de calcul de la marche des rayons.

II.3 Conception de Ray Marching

Tout d'abord, la scène 3D à rendre doit être définie. Il s'agit des éléments qui composent la scène, tels que les objets, les lumières et les ombrages. Les objets sont souvent définis implicitement, c'est-à-dire par des formules mathématiques.

Projection de rayons lumineux : Le processus de marche par rayons commence par l'envoi de rayons lumineux depuis l'œil de la caméra vers la scène virtuelle. Pour chaque pixel de l'image finale, un rayon est

tiré depuis la position de la caméra à travers ce pixel.

Initialisation du rayon : Lorsqu'un rayon est lancé, il doit être initialisé avec des informations de base telles que son origine (position de la caméra) et sa direction (le vecteur reliant la caméra au pixel de l'image).

Boucle de ray marching : À partir de l'origine du rayon, vous entrez dans une boucle de ray marching qui se répète jusqu'à ce que certaines conditions de sortie soient remplies. Cette boucle itère le long du rayon en échantillonnant les points à intervalles réguliers.

Distance signée : À chaque itération de la boucle de ray marching, vous devez calculer la distance signée entre le point actuel sur le rayon et l'objet le plus proche dans la scène. La distance signée est une mesure de la distance entre le point et la surface de l'objet, et elle peut être :

Positive (à l'extérieur de l'objet).

Négative (à l'intérieur de l'objet).

Nulle (sur la surface de l'objet).

Condition de sortie : La boucle de ray marching se termine lorsque certaines conditions sont remplies. Par exemple : une limite à la distance maximale qu'un rayon peut parcourir. une tolérance pour une distance signée (si elle est très proche de zéro, le point peut être considéré comme étant à la surface de l'objet).

Calcul de la couleur : Une fois que la boucle de ray marching est terminée, vous avez atteint un point sur la surface de l'objet. À ce stage, la couleur de ce point est calculée en fonction de diverses propriétés, telles que les lumières et les ombres présentes dans la scène.

Échantillonnage multiple : Pour améliorer la qualité de l'image finale.

Génération de l'image finale : Une fois que vous avez calculé la couleur pour chaque pixel de l'image en lançant les rayons correspondants, vous pouvez générer l'image finale en assemblant tous les pixels ensemble.

II.4 Version CPU

La marche des rayons peut être implémentée à la fois sur le CPU et sur le GPU, chacun ayant ses propres avantages et inconvénients.

Sur le CPU, la marche des rayons est généralement implémentée à l'aide d'un algorithme série, où chaque pixel est traité séquentiellement, un à la fois. Bien que cette approche puisse être plus lente que le traitement parallèle sur un GPU, elle a l'avantage d'être plus flexible et personnalisable, ainsi que de pouvoir fonctionner sur une plus large gamme de matériel. De plus, comme le CPU a accès à plus de mémoire que le GPU, il peut gérer des scènes plus grandes et plus complexes, bien qu'à une fréquence d'images plus lente.

Bien que la mise en œuvre de la marche des rayons sur un CPU [12] puisse être plus difficile que sur un GPU, elle a l'avantage d'être plus accessible aux débutants et peut être un bon point de départ pour en apprendre davantage sur la marche des rayons et l'infographie en général. De plus, en optimisant l'algorithme et en utilisant des techniques de traitement parallèle, il est possible d'obtenir un rendu en temps réel sur un processeur, bien que cela puisse nécessiter une optimisation et des ressources matérielles importantes.

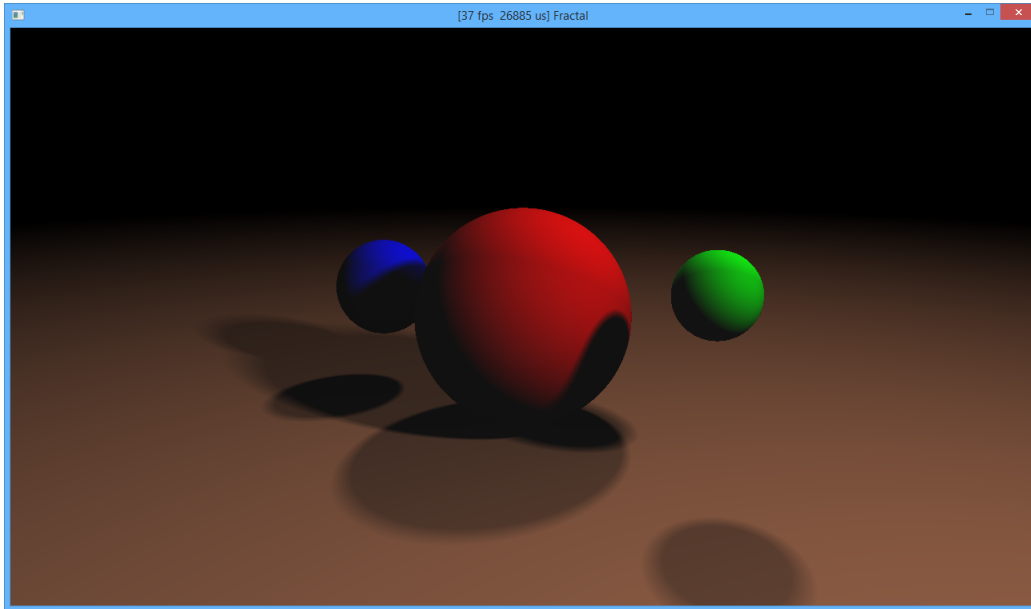


FIGURE II.2 – La Version CPU de Ray Marching [32]

II.5 Version GPU

La marche des rayons est particulièrement bien adaptée à une implémentation sur un GPU en raison de sa capacité à effectuer des calculs massivement parallèles [16]. Les GPU ont un grand nombre de cœurs de traitement, qui peuvent être utilisés pour calculer simultanément l'intersection de plusieurs rayons avec les objets de la scène. Cela permet un rendu en temps réel de scènes complexes avec des fréquences d'images élevées et des niveaux de détail élevés.

Bien que la mise en œuvre sur un GPU puisse être plus complexe que sur un CPU, elle présente plusieurs avantages. Premièrement, les GPU sont conçus spécifiquement pour le calcul parallèle et peuvent gérer un nombre beaucoup plus important de calculs en parallèle qu'un CPU. Cela signifie que les algorithmes de ray marching peuvent être optimisés pour s'exécuter beaucoup plus rapidement sur un GPU que sur un CPU. Deuxièmement, les GPU sont optimisés pour le traitement graphique et disposent d'un matériel spécialisé pour des tâches telles que le mappage de texture et la rasterisation, ce qui peut encore améliorer les performances.

Dans l'ensemble, la mise en œuvre de la marche des rayons sur un GPU est un moyen puissant et efficace de rendre des scènes complexes avec des niveaux élevés de détails et d'interactivité [33], ce qui en fait un choix populaire pour de nombreuses applications d'infographie, de jeux et de visualisation scientifique.

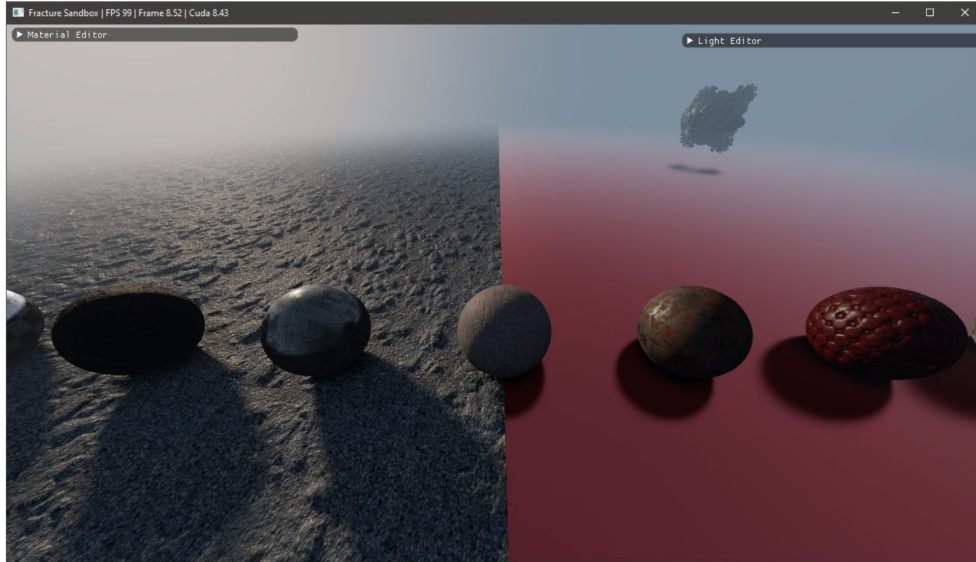


FIGURE II.3 – La Version GPU de Ray Marching [16]

II.6 La Différence entre la version CPU et la version GPU

La différence entre la version CPU et la version GPU du ray MARCHING réside principalement dans la manière dont les calculs sont effectués.

Version CPU	Version GPU
les calculs sont effectués séquentiellement par le processeur central. les calculs sont lents pour des scènes complexes ou avec de nombreuses sources de lumière. les processeurs sont très efficaces pour effectuer des tâches plus générales.	les cartes graphiques (GPU) sont conçues pour effectuer des calculs en parallèle, ce qui les rend beaucoup plus rapides pour les calculs complexes tels que le ray MARCHING.
La version GPU du raymarching est plus rapide et plus efficace pour traiter des scènes 3D complexes en temps réel.	la version CPU est mieux adaptée aux tâches plus générales nécessitant une puissance de calcul plus élevée et une plus grande flexibilité.

TABLE II.1 – La Différence entre la version CPU et la version GPU

II.7 Les travaux connexe

II.7.1 Ray Marching and Signed Distance Functions

Jamie Wong [38] est une développeuse de logiciels et une artiste qui a travaillé sur plusieurs projets impliquant le raymarching. Elle a notamment créé une bibliothèque open-source appelée "Distance Field

Font Rendering” qui utilise le raymarching pour générer des polices de caractères en temps réel.

En général, Jamie Wong semble être une adepte du raymarching et de ses capacités en matière de rendu en temps réel. Elle a écrit plusieurs articles de blog et tutoriels sur cette technique et a également présenté des conférences sur le sujet.

Dans un de ses articles, elle explique que le raymarching permet d’obtenir des images de haute qualité avec un nombre de polygones relativement faible, ce qui peut être utile dans les environnements en temps réel tels que les jeux vidéo et la réalité virtuelle. Elle mentionne également que cette technique peut être utilisée pour générer des effets visuels complexes tels que des fractales, des effets de lumière et de matière.

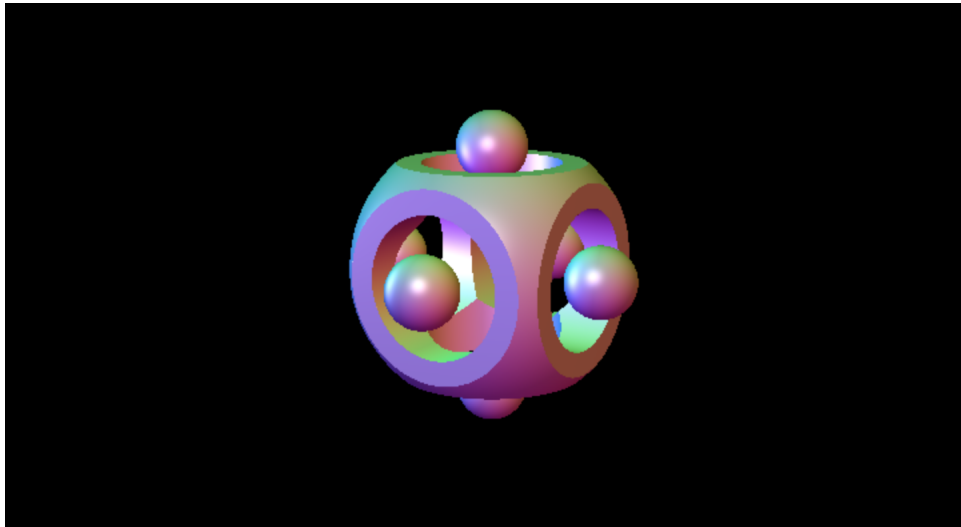


FIGURE II.4 – Conception de Jamie wong [38]

Jamie Wong semble être une partisane enthousiaste du raymarching et de ses capacités en matière de rendu en temps réel.

Elle a travaillé sur des projets utilisant cette technique et a écrit des articles de blog et des tutoriels pour aider les autres à apprendre à l’utiliser.

II.7.2 Ray Marching de Michael walczyk

lorsque nous utilisons la marche des rayons, nous n’avons plus affaire à des polygones. Mais si nous n’avons pas de points, de lignes, de triangles et de maillages, comment rendre quoi que ce soit ? L’astuce consiste à utiliser des fonctions de distance signées ou SDF . Ce sont des fonctions mathématiques qui prennent un point dans l’espace et vous indiquent à quelle distance ce point se trouve d’une surface.

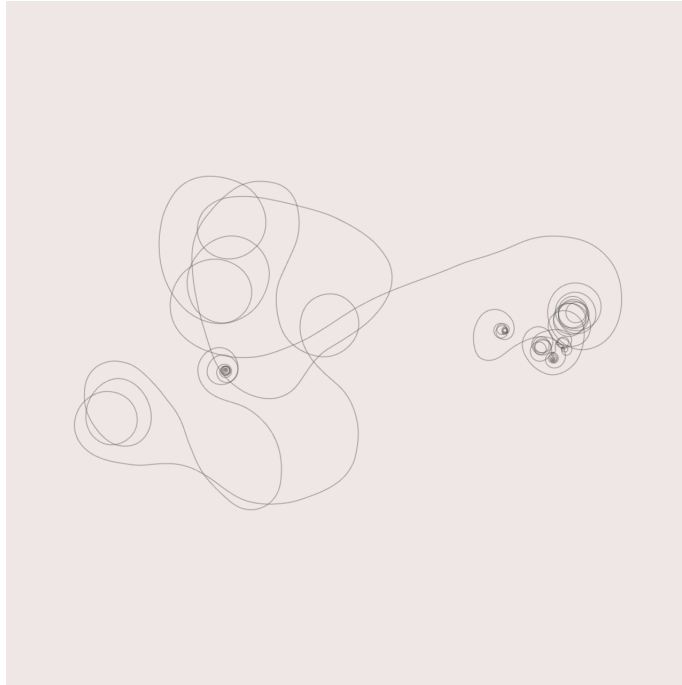


FIGURE II.5 – La Conception de Michael Walczyk [35]

Pour une sphère, la normale en tout point de la surface peut être calculée en normalisant simplement le vecteur $c - p$, où comme précédemment, c est le centre de la sphère et p est un point sur la surface de la sphère. [35]

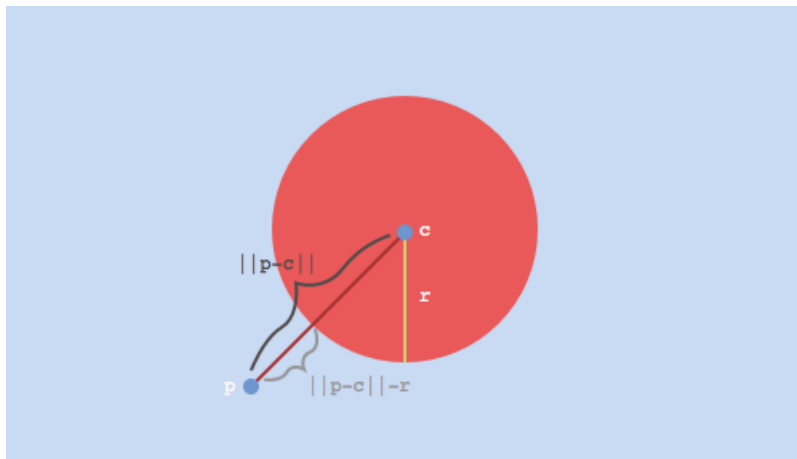


FIGURE II.6 – Circle-Distance [35]

Génération de rayons : N'oubliez pas que tout notre code s'exécutera à l'intérieur d'un seul fragment shader, il y a donc un petit saut, où nous devons en quelque sorte décrire un monde 3D à partir du plan 2D sur lequel notre fragment shader s'exécute. Un shader de fragment s'exécute une fois pour chaque pixel qui constitue notre image finale rendue.

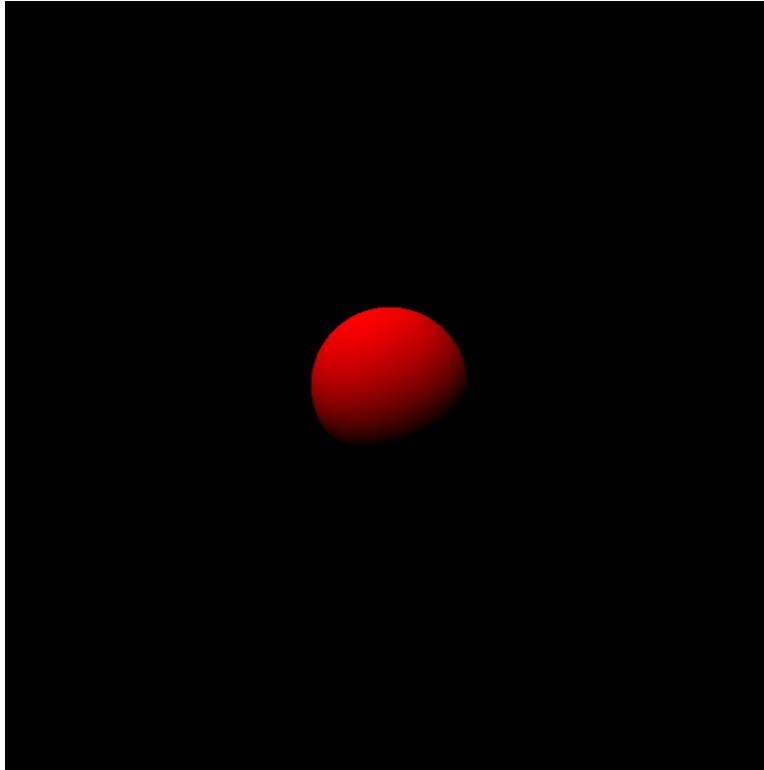


FIGURE II.7 – La Génération de Michael Walczyk [35]

Déformer la fonction de distance : Une fois que nous avons une configuration de marche des rayons de base, nous pouvons « pousser » ou perturber nos fonctions de distance pour créer des formes plus intéressantes.

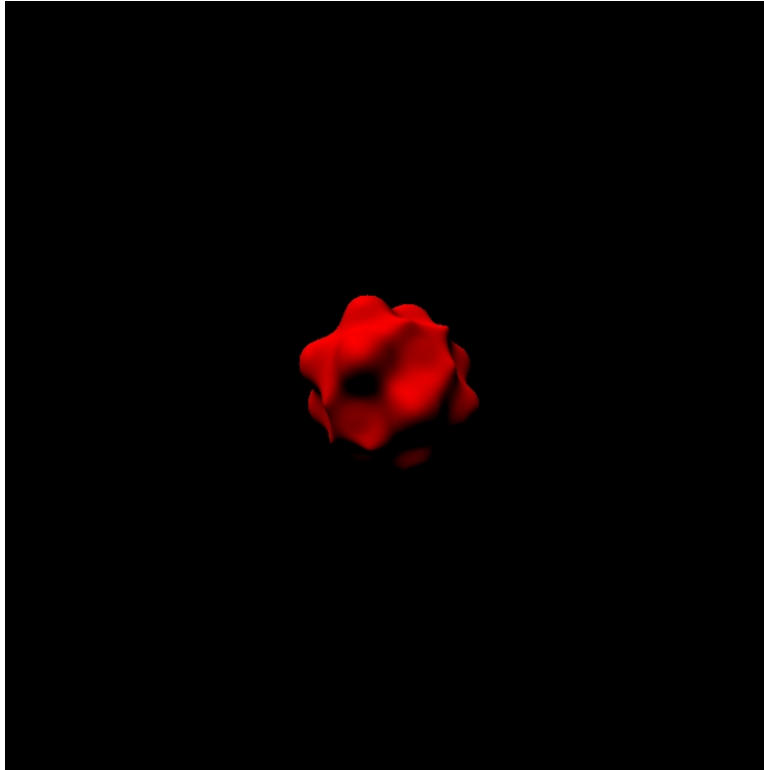


FIGURE II.8 – Déformer la fonction de distance [35]

II.7.3 Une boîte à outils d’unité hautement interactive pour la construction visuelle de champs de distance signés

Kevin Watters et Fernando Ramallo [24] sont deux développeurs de jeux vidéo indépendants qui ont travaillé sur des projets de jeux utilisant le raymarching. Leurs opinions sur cette technique de rendu sont les suivantes :

Kevin Watters a travaillé sur le jeu "Igneos : The Last Phoenix", qui utilise le raymarching pour créer des paysages 3D détaillés et immersifs. Dans une interview avec le site web Gamasutra, il a déclaré que le raymarching est une technique de rendu très puissante, mais qui nécessite des connaissances avancées en mathématiques et en programmation pour être mise en œuvre correctement. Il a également noté que le raymarching peut être gourmand en ressources, ce qui peut poser des problèmes de performances dans les jeux vidéo.

Fernando Ramallo a travaillé sur le jeu "Panoramical", qui utilise également le raymarching pour créer des paysages et des environnements immersifs. Dans une interview avec le site web The Creators Project, il a décrit le raymarching comme une technique de rendu "magique", qui permet aux artistes de créer des effets de lumière et de transparence incroyablement réalistes. Cependant, il a également noté que le raymarching peut être difficile à mettre en œuvre et qu’il peut nécessiter des ajustements de performances pour fonctionner correctement sur différentes plates-formes.

Kevin Watters et Fernando Ramallo semblent considérer le raymarching comme une technique de rendu puissante et expressive, mais qui peut être difficile à mettre en œuvre et peut poser des problèmes de performances dans les jeux vidéo.

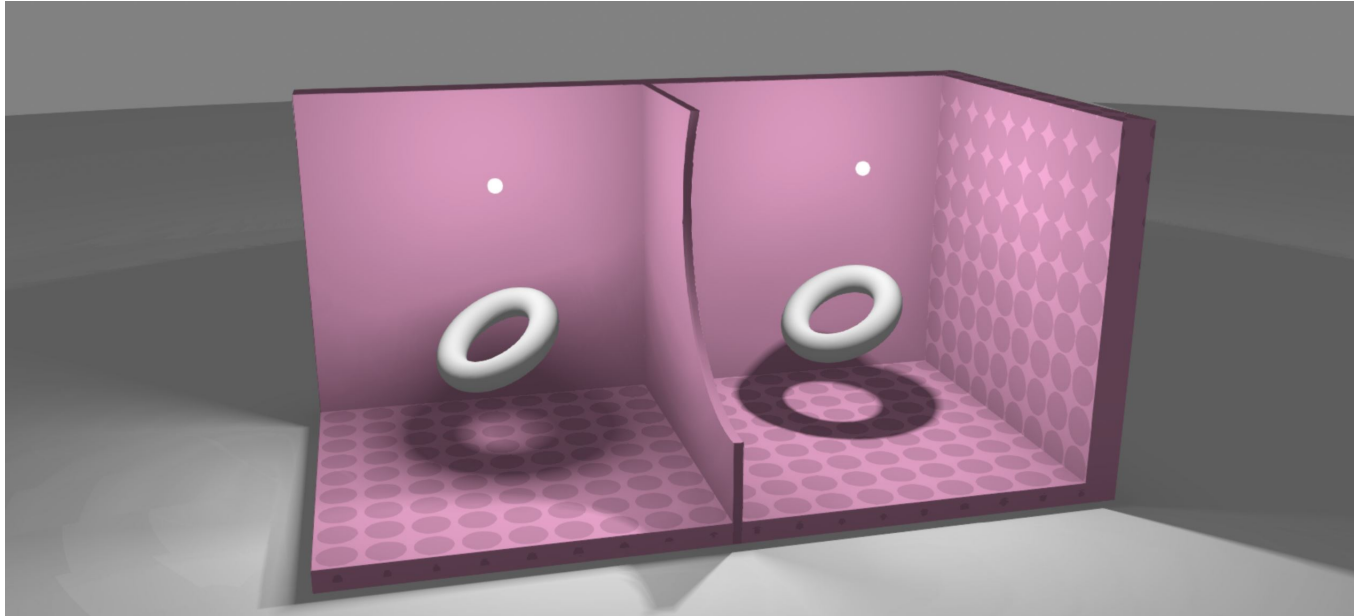


FIGURE II.9 – Boîte à outils de Raymarching pour Unity [36]

II.7.4 Ray Marching de Tyler Bovenzi

Ray Marching est une technique de rendu très différente qui n'utilise pas de polygones, de panneaux d'affichage ou de maillages.

Il utilise plutôt des fonctions de distance signées pour calculer mathématiquement la couleur de chaque pixel.

L'une de ses conclusions est que bien qu'il soit coûteux en calcul par rapport aux techniques de rendu traditionnelles, il offre certains avantages et des propriétés intéressantes.

La Génération Ray : Pour chaque pixel rendu, le fragment shader doit calculer un vecteur à partir de la caméra pour couvrir un certain azimuth vertical et horizontal dans la vue. [9]

Ces azimuths créent ce que l'on appelle traditionnellement le champ de vision. Dans le schéma ci-dessous, le vecteur d'un pixel est en rouge tandis que le champ de vision est délimité par des lignes vertes.

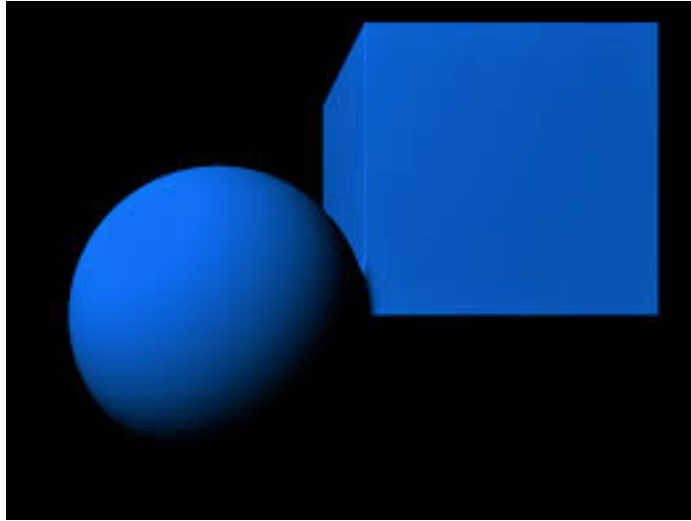


FIGURE II.10 – Ombrage des Distances [9]

II.7.5 Ray Marching Distance Fields with CUDA

Avelina Hadji-Kyriacou et Ognjen Arandjelovic [17] sont deux chercheurs en informatique à l'Université d'Oxford, qui ont co-écrit un article de recherche sur le raymarching en 2020. Dans cet article, ils ont examiné l'utilisation du raymarching pour la génération de nuages en temps réel dans les jeux vidéo.

Bien que je ne puisse pas connaître leurs opinions personnelles sur le raymarching en dehors de leur recherche, leur article montre qu'ils considèrent le raymarching comme une technique de rendu prometteuse pour la génération de nuages en temps réel dans les jeux vidéo.

Ils soulignent que le raymarching permet de générer des nuages réalistes avec des ombres et des effets de lumière naturels, ce qui peut améliorer considérablement l'expérience de jeu pour les joueurs. Cependant, ils notent également que le raymarching est gourmand en ressources et nécessite une puissance de traitement importante pour générer des nuages en temps réel.

Dans l'ensemble, Hadji-Kyriacou et Arandjelovic semblent considérer le raymarching comme une technique de rendu puissante, mais qui nécessite des ressources importantes pour être mise en œuvre. Leur recherche montre également que le raymarching peut avoir des applications pratiques dans l'industrie du jeu vidéo.

II.8 Conclusion

Le Ray Marching est une technique puissante en infographie qui présente de nombreux avantages. Tout d'abord, il offre une grande flexibilité en permettant le rendu de scènes complexes et détaillées avec des objets géométriques sophistiqués. Il peut prendre en charge une variété de formes, y compris des surfaces lisses, des fractales et des surfaces implicites, sans nécessiter de représentations géométriques explicites.

De plus, le raymarching permet d'obtenir un réalisme élevé en projetant des rayons dans une scène et en calculant avec précision les intersections avec les objets. Cette précision contribue à créer des effets d'éclairage

et d'ombre réalistes, ce qui rend les scènes visuellement attrayantes et immersives.

Un autre avantage du raymarching est son efficacité. Il s'agit d'une technique de rendu en temps réel qui permet d'obtenir des résultats de haute qualité tout en conservant un coût de calcul relativement faible. Sa nature itérative et sa capacité à effectuer des calculs de manière progressive facilitent un rendu efficace, surtout lorsqu'il est combiné avec des structures d'accélération telles que des hiérarchies de volumes englobants (BVH) ou des champs de distance.

En outre, le raymarching ouvre de nouvelles possibilités créatives pour les artistes et les designers. Il permet la génération de scènes procédurales complexes, de paysages fractals et de visualisations abstraites, favorisant ainsi l'exploration et l'expérimentation artistiques.

Cependant, malgré ses nombreux avantages, le raymarching présente également certaines limites. Il peut être gourmand en termes de calcul pour des scènes très complexes ou contenant un grand nombre d'objets. De plus, l'obtention d'effets tels que des réflexions précises et un éclairage global peut être difficile avec le raymarching seul. Néanmoins, grâce aux progrès matériels et aux techniques de rendu en constante évolution, ces limitations sont en cours de résolution, ce qui fait du raymarching un outil de plus en plus populaire et puissant en infographie.

Chapitre III

Mise en œuvre et résultats

Mise en œuvre et résultats

III.1 Introduction

Pour concrétiser notre travail, il faut d'abord créer un modèle qui permette de formaliser les premières étapes de développement de notre technique sur le GPU pour trouver une solution viable et répondre aux exigences.

Ce chapitre est consacré à la présentation de la conception, de la mise en œuvre et des résultats obtenus par la technique. La première partie consiste à introduire la structure générale de notre système. La deuxième partie décrit les détails de mise en œuvre. La dernière partie décrit les résultats.

III.2 Environnements et outils

Cette section décrit les détails de l'implémentation de notre technique. Nous vous présentons l'environnement et les outils logiciels et matériels nécessaires pour la réalisation de notre travail.

III.2.1 la première partie

Pour atteindre l'objectif de notre travail, nous avons besoin de différents environnements et outils : langages de programmation, API, des bibliothèques, des IDE, etc.

III.2.1.1 Matériel de configuration

- Pc portable : MSI
- Processeur : intel(R) core(TM) i5-7300HQ CPU@2.50 GHz 2.50GHz.
- La carte graphique : GeForce GTX1060.
- Systeme d'exploitation : Windows 10, 64 bits.
- API graphique : OpenGL 4.5.
- GLSL (The OpenGL Shading Language).

III.2.2 la deuxième partie

III.2.2.1 OpenGL GLSL

Le langage d'ombrage OpenGL (GLSL) est le principal langage d'ombrage pour OpenGL. Alors que, grâce aux extensions OpenGL, il existe plusieurs langages d'ombrage disponibles pour une utilisation dans OpenGL, GLSL (et SPIR-V) sont pris en charge directement par OpenGL sans extensions.

GLSL est un langage de style C. Le langage a subi un certain nombre de changements de version et partage le modèle de dépréciation d'OpenGL. La version actuelle de GLSL est la 4.60.

III.3 Architecture générale

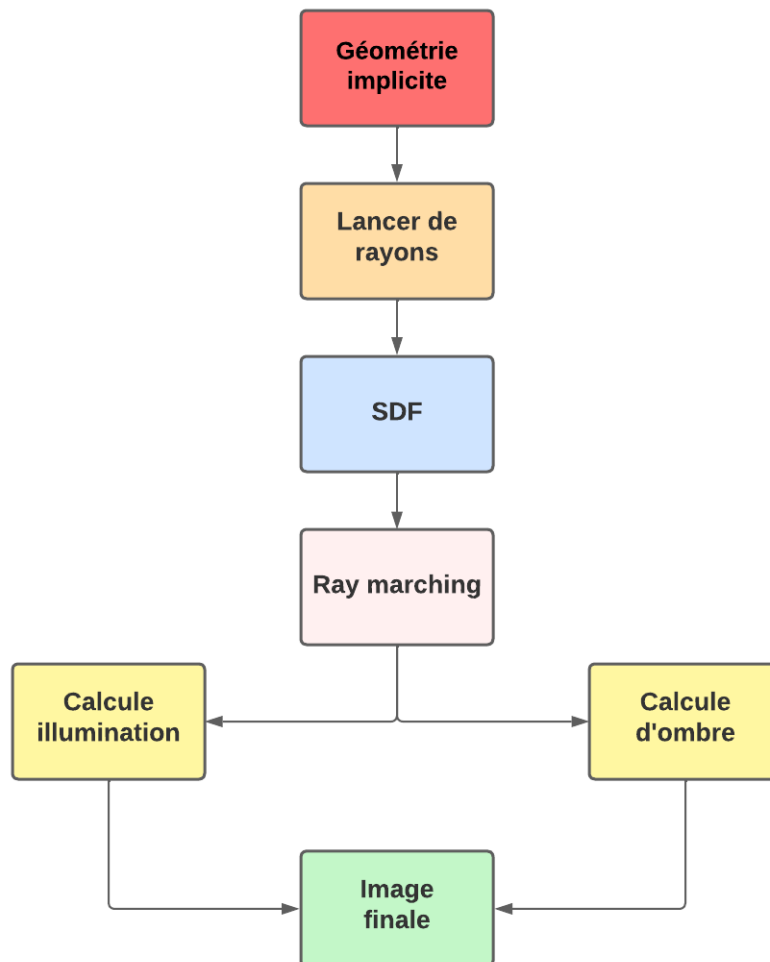


FIGURE III.1 – L'architecture Générale du Système

III.3.1 La géométrie implicite

Dans la géométrie implicite, les objets géométriques sont décrits par des équations de la forme $F(x, y, z) = 0$, où F est une fonction polynomiale ou transcendantale.

Au lieu d'exprimer explicitement les coordonnées x , y et z d'une sphère en fonction de paramètres, la géométrie implicite se concentre sur les relations entre les coordonnées qui définissent la sphère.

L'équation implicite générale d'une sphère en géométrie tridimensionnelle est :

$$F(x,y,z) = (x - a)^2 + (y - b)^2 + (z - c)^2 - r^2 = 0$$

- (a,b,c) sont les coordonnées du centre de la sphère.
- r est rayon de la sphère (radius).

Chaque point (x, y, z) satisfaisant cette équation est à une distance r du centre de la sphère.

III.3.2 Lancer de rayons

Une scène virtuelle doit être définie en spécifiant la géométrie de l'objet, les matériaux, l'éclairage et d'autres propriétés. Chaque rayon émane de la caméra et traverse la scène.

Si le matériau de l'objet est réfléchissant ou réfractant, un rayon réfléchi ou réfracté est généré et émis dans la scène.

Pour déterminer si un point de la scène est éclairé, projetez un rayon d'ombre depuis ce point vers la source de lumière.

Lorsque tous les rayons ont fini d'interagir avec les objets de la scène, les informations de couleur et d'ombre sont collectées et utilisées pour déterminer la couleur finale de chaque pixel de l'image.

III.3.3 La fonction de distance signée (SDF)

Après Représentation d'objet et avantages, la fonction qui prend un point dans l'espace 3D en entrée et renvoie la distance signée à la surface de l'objet. La fonction doit prendre en compte la représentation de l'objet et calculer la distance en conséquence.

À l'intérieur de l'objet, la distance est négative.

À l'extérieur de l'objet, la distance est positive.

À la surface de l'objet, la distance est nulle.

La distance est calculée en trouvant le point le plus proche sur la surface de l'objet.

Les distances signées calculées peuvent être stockées dans une structure de données pour un accès efficace pendant le rendu ou la détection de collision.

Il est important de noter que les détails d'implémentation spécifiques de la fonction de distance signée dépendront de la représentation d'objet choisie.

Estimer la normale

Choisissez un point dans l'espace 3D où vous souhaitez estimer la normale.

Connectez le point choisi le long de chaque axe (x, y et z) avec une petite distance epsilon pour obtenir trois points adjacents Obtenez des valeurs de distance signées.

Calcul du gradient SDF.

Normaliser le vecteur de gradient calculé pour obtenir la normale de surface estimée.

III.3.4 Algorithme de Ray Marching

Dans l'algorithme plusieurs étapes :

-Déterminez la position et l'orientation de la caméra dans la scène 3D.

-Renvoie la distance jusqu'à la surface la plus proche atteinte pendant le flux de rayons.

-Insérer une boucle dans le Ray Marching.

-Déplacez le point de départ du rayon d'une distance fixe le long de sa direction. Calculez la distance jusqu'à la surface la plus proche à ce nouveau point de départ à l'aide de la fonction de distance signée (SDF) ou d'une autre méthode de détection de collision.

-Vous pouvez calculer la luminance et la couleur d'un pixel en fonction de sa position sur la surface, des propriétés du matériau et des sources de lumière dans la scène.

III.3.5 BRDF

Des modèles BRDF plus avancés, tels que des modèles d'ombrage basés sur la physique, sont souvent utilisés pour obtenir des résultats de rendu plus précis et réalistes.

le modèle d'ombrage Blinn-Phong

Calculez la direction de la lumière.

Calculez la direction de la vue.

Normaliser la direction de la lumière et la direction de la vue.

Calculez la normale à la surface.

Calculez le terme spéculaire et diffuse.

Combinez les termes diffus et spéculaire pour obtenir le résultat d'ombrage final.

III.3.6 Calcule d'illumination et l'ombre

Calcule d'illumination

La formule générale pour calculer l'éclairage d'un point en utilisant le modèle Phong-Blinn est la suivante :

$$I = I_a * K_a + I_p * K_d * (N \cdot L) + I_p * K_s * (H \cdot N) s$$

Calcule d'ombre

Pour chaque objet de la scène, on calcule la position de son ombre en fonction de la position de la source de lumière. On projette ensuite cette ombre sur la surface sur laquelle elle tombe, en utilisant les coordonnées de la source de lumière et de l'objet.

On peut utiliser différentes techniques pour représenter l'ombre, comme les ombres dures, les ombres douces.

Cette méthode est relativement rapide, mais elle ne tient pas compte des effets de la lumière indirecte ou de la transmission de la lumière à travers les objets transparents.

Outre la précision, l'opacité de l'ombre est calculée en fonction de la distance parcourue le long du rayon et de la distance jusqu'à la surface la plus proche.

Le résultat du vignettage peut varier en fonction de plusieurs facteurs tels que la distance maximale délibérée, le coefficient d'atténuation.

III.3.7 Scène finale

L'affichage des résultats finale avec un rendu complet.

III.4 Description détaillé du système

III.4.1 Les Fichiers Shaders

shader1.vert

Ce fichier contient les vertex shaders. Il est utilisé pour manipuler les sommets des objets dans la scène 3D.

shader1.frag

Ce fichier contient un volet de shader. Il est utilisé pour spécifier la couleur de pixel finale d'un objet lors du rendu.

Ils incluent l'éclairage, les fonctions et les calculs pour la technologie Ray marching.

III.4.2 La fonction de distance signée

III.4.2.1 La fonction de distance signée d'une sphère

Cette fonction prend trois paramètres d'entrée :

- * la position dans l'espace 3D pour laquelle le SDF est calculé.
- * le rayon de la sphère.
- * la position centrale de la sphère.

La fonction calcule le SDF en prenant (la longueur du vecteur position + centre et en soustrayant le rayon de la sphère). La fonction `length()` calcule la distance euclidienne entre l'origine (0, 0, 0) et le vecteur donné.

Le résultat de ce calcul représente la distance signée entre la position et la surface de la sphère.

- Si le résultat est négatif, cela signifie que la position est à l'intérieur de la sphère.
- Si le résultat est positif, cela signifie que la position est en dehors de la sphère.
- si le résultat est zéro, cela signifie que la position est exactement à la surface de la sphère.

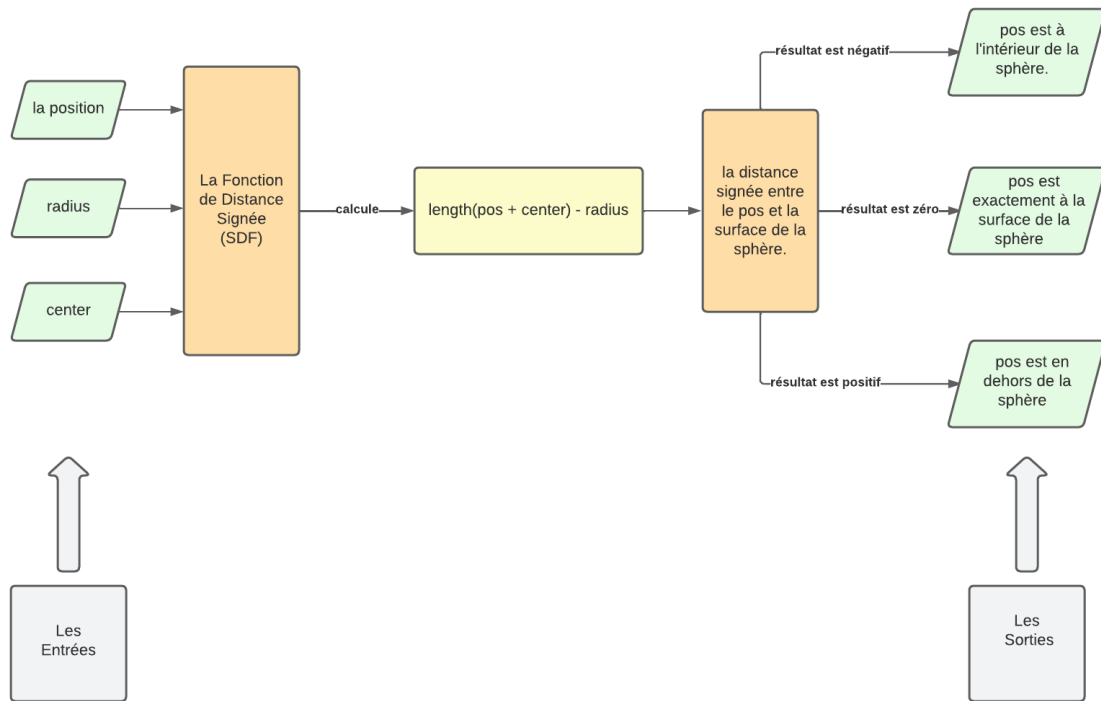


FIGURE III.2 – Diagramme de La fonction de distance signée d’une sphère

III.4.2.2 La fonction de distance signée d’une plan

Cette fonction prend deux paramètres d’entrée :

- * la position dans l’espace 3D pour laquelle le SDF est calculé.
- * un vecteur à quatre composants représentant la normale et la distance du plan.

Le calcul du SDF est simple. Il calcule le produit scalaire entre le vecteur de position et le vecteur normal, qui donne la distance de position au plan le long de la direction de la normale du plan. Il ajoute ensuite la composante w normale pour tenir compte du décalage de distance signé du plan.

La valeur résultante représente la distance signée entre le position et le plan.

Si la valeur est négative, cela signifie que la position est en dessous du plan.

Si la valeur est positive, cela signifie que la position est au-dessus du plan.

si la valeur est zéro, cela signifie que la position est exactement sur le plan.

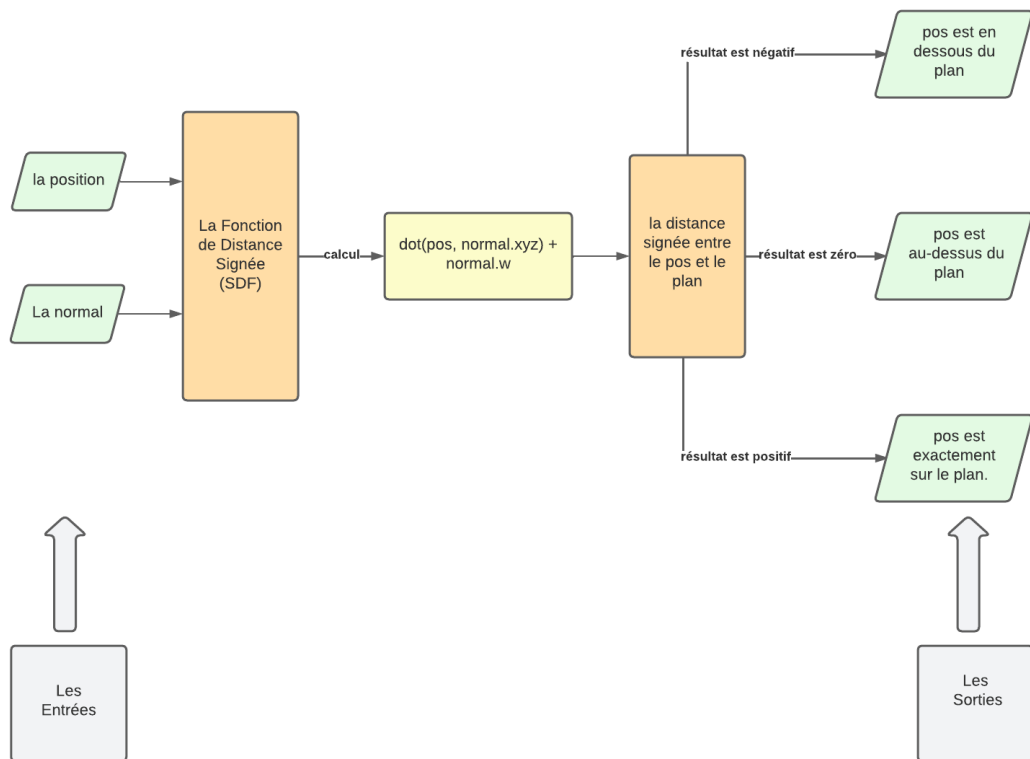


FIGURE III.3 – Diagramme de La fonction de distance signée d’un plane

III.4.2.3 La fonction de distance signée (SDF) d’une sphère et d’un plan

La fonction prend une position dans l’espace 3D en entrée et renvoie la distance signée à l’objet le plus proche dans la scène. Il attribue également un ID de collision pour indiquer quel objet est le plus proche.

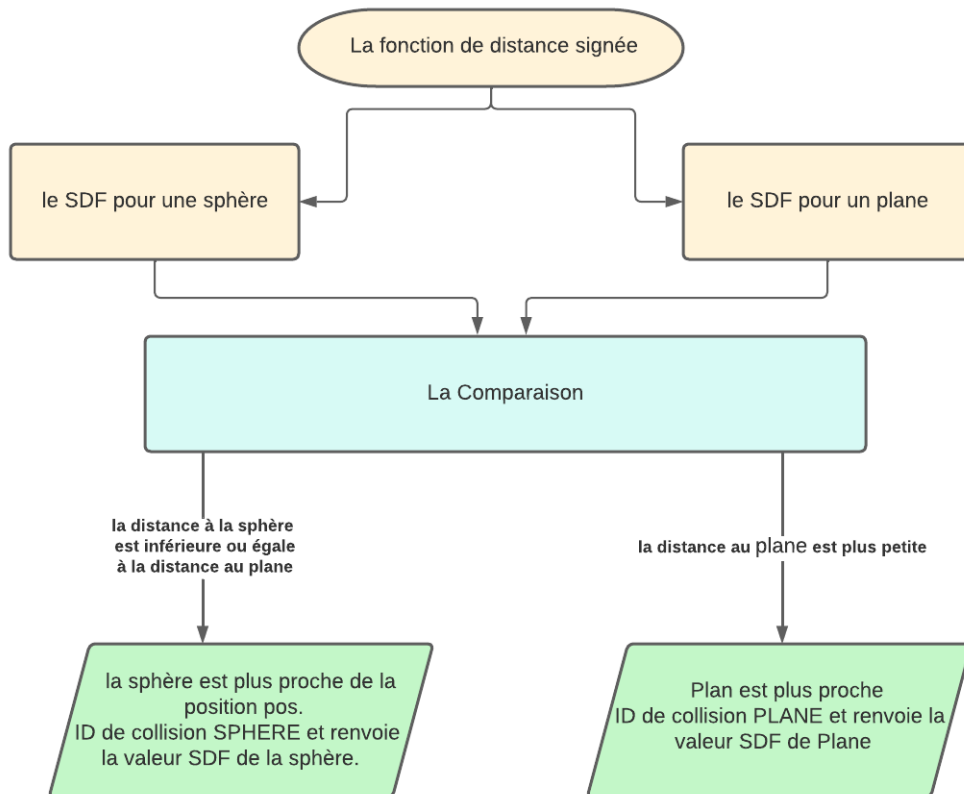


FIGURE III.4 – Diagramme de La fonction de distance signée d’un plane et une sphère

III.4.2.4 La fonction estimateNormal

La fonction prend une position 3D en entrée et renvoie la normale de surface estimée à cette position.

Pour estimer la normale, la fonction évalue la fonction de distance signée de la scène (sceneSDF) à six points voisins.

En soustrayant les valeurs de distance signées aux points voisins, la fonction se rapproche du taux de variation de la fonction de distance dans chaque direction de coordonnées. Ces différences de distances se rapprochent des composants de la normale de surface.

Le vecteur résultant est ensuite normalisé pour s’assurer que la normale estimée a une longueur de 1, produisant un vecteur unitaire qui représente la normale de surface au point.

III.4.3 La Fonction de distribution bidirectionnelle de la réflectance

III.4.3.1 La modèle d'éclairage Blinn-Phong

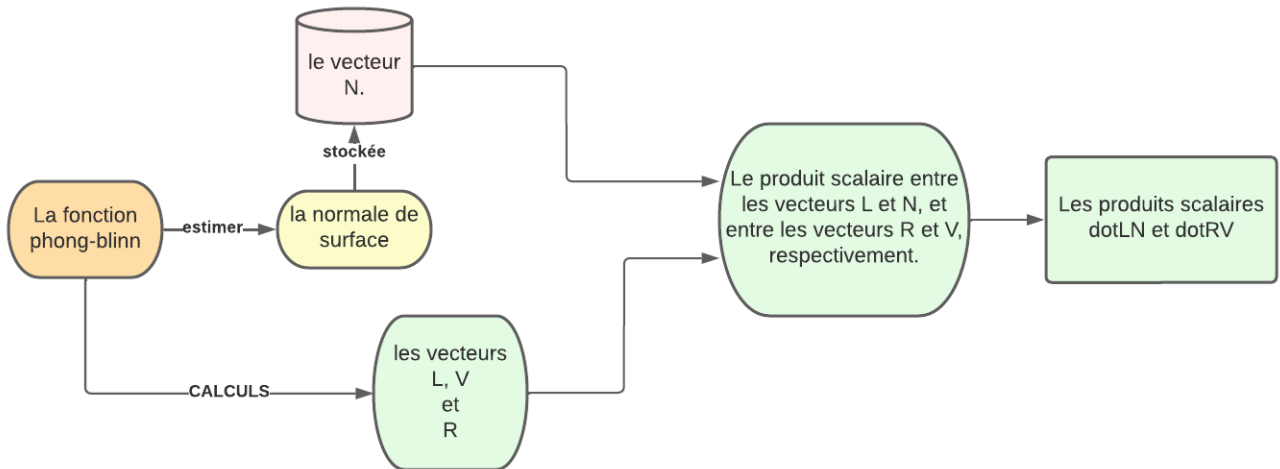


FIGURE III.5 – Diagramme de contribution à la lumière

Si le produit scalaire (dotLN) est inférieur à zéro, cela signifie que la lumière n'est pas visible depuis ce point sur la surface. Dans ce cas, la contribution de la lumière est nulle.

Si le produit scalaire (dotRV) est inférieur à zéro, cela signifie que la réflexion de la lumière est dans la direction opposée à celle de l'œil (la caméra). Dans ce cas, seule la composante diffuse de l'éclairage est calculée et retournée en multipliant la couleur diffuse par le produit scalaire dotLN , puis en multipliant par l'intensité de la lumière et l'atténuation.

Pour les autres cas, le calcul Blinn-Phong est effectué. Le vecteur half direction est calculé en normalisant la somme du vecteur L et du vecteur V .

Ensuite, la composante spéculaire est calculée en utilisant la formule Blinn-Phong, en prenant le produit scalaire entre half direction et N , puis en élevant au carré le résultat et en le multipliant par l'intensité de la lumière spéculaire, le facteur d'atténuation et la puissance spéculaire (α).

La contribution totale de la lumière est calculée en combinant la composante diffuse et la composante spéculaire, en multipliant chaque terme par l'intensité de la lumière, la couleur diffuse, le produit scalaire dotLN et l'atténuation.

III.4.3.2 L'illumination d'un point donné sur une surface

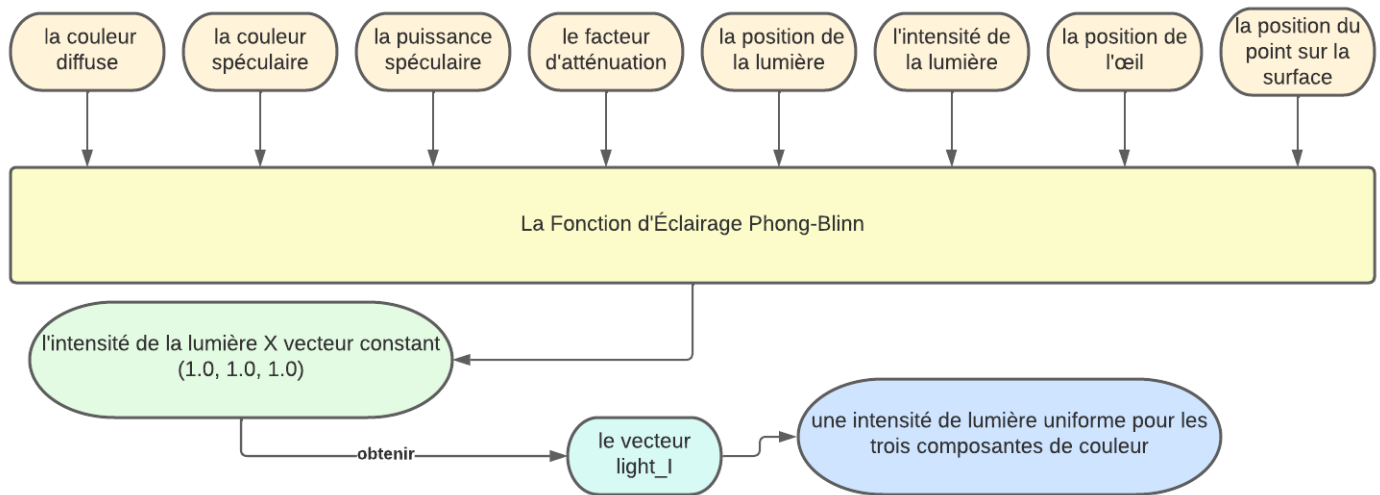


FIGURE III.6 – Diagramme de La Fonction d'Éclairage Phong-Blinn

La fonction calcule la contribution de la lumière au point donné sur la surface. Les paramètres passés à cette fonction sont :

la couleur diffuse, la couleur spéculaire, la puissance spéculaire, la position du point, la position de l'œil, la position de la lumière, l'intensité de la lumière et le facteur d'atténuation.

La couleur calculée est retournée comme résultat.

III.4.3.3 La direction d'un rayon

La fonction prend en entrée :

Le champ de vision, la taille de l'écran et les coordonnées du fragment.

La direction du rayon est calculée en créant un vecteur à trois dimensions avec les coordonnées xy du fragment et la coordonnée -z pour la profondeur. Le vecteur est normalisé pour obtenir une direction unitaire.

III.4.4 Les calculés d'Algorithme de Ray Marching

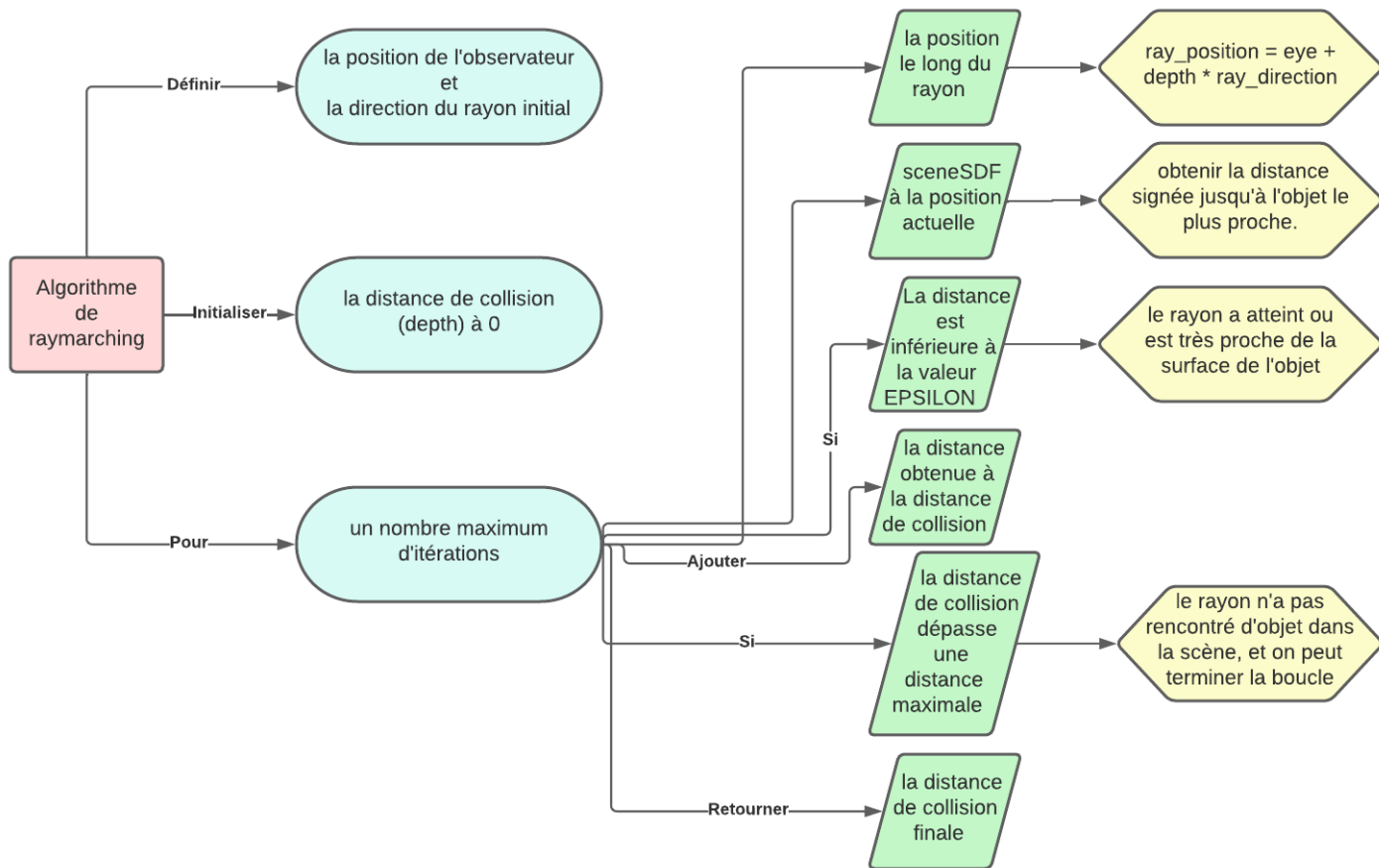


FIGURE III.7 – Diagramme de L'Algorithme de Ray Marching

L'algorithme de marche des rayons itère le long du rayon à l'aide de la fonction de distance signée de la scène, en estimant la distance à l'objet le plus proche à chaque étape. Cumulez la distance parcourue (profondeur) pour déterminer la distance totale parcourue par le faisceau avant d'entrer en collision ou de respecter la condition de terminaison.

la scène doit être décrite par une fonction de distance signée qui renvoie la distance à l'objet le plus proche à un emplacement donné, ainsi que d'autres informations pertinentes (identifiants des objets en collision, etc.).

III.4.5 Les paramètres de couleur damier

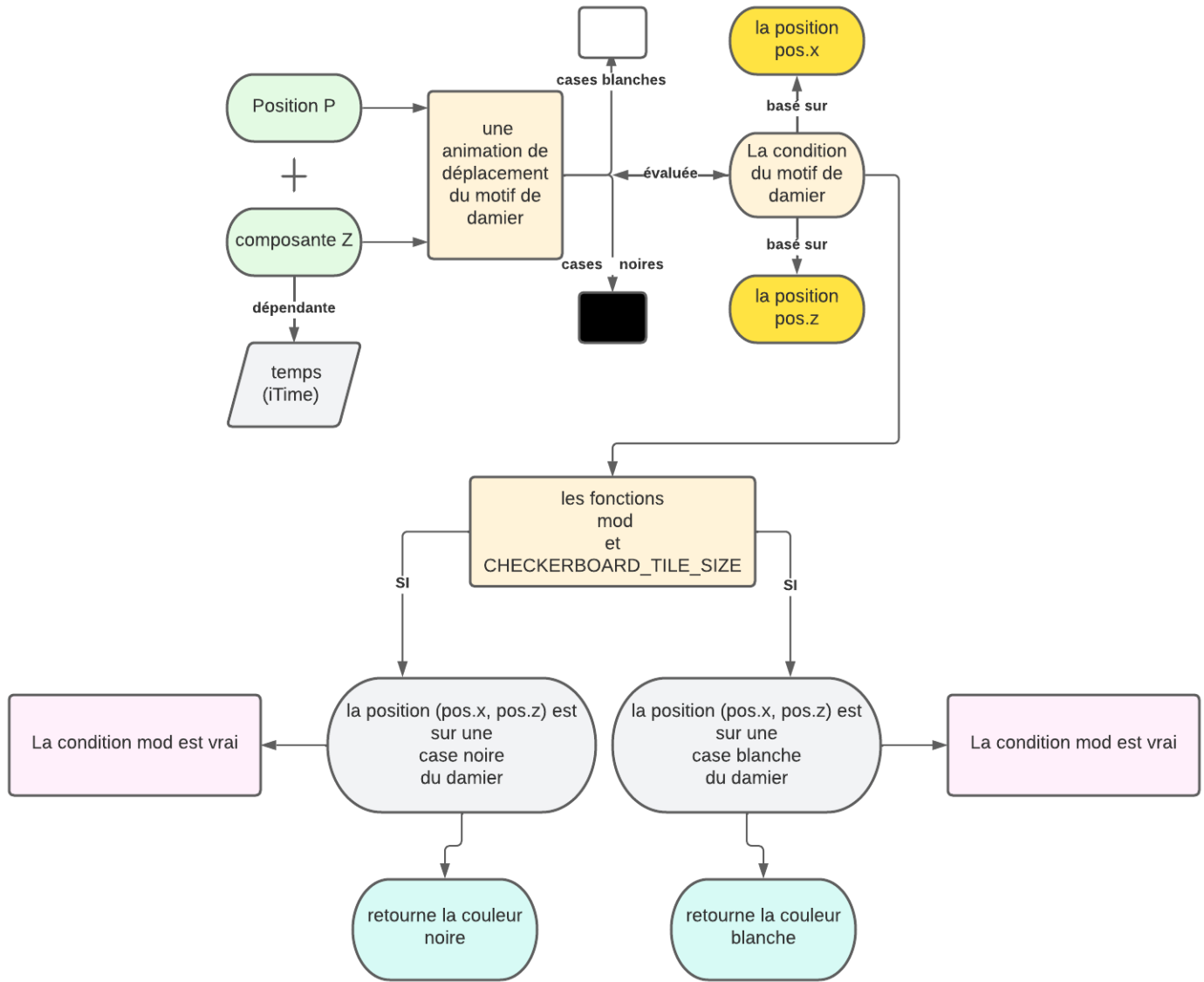


FIGURE III.8 – Diagramme de damier

La fonction est utilisée pour générer un motif de damier sur une surface en fonction de la position dans l'espace. La variation de la position Z en fonction du temps donne un effet d'animation au motif de damier.

III.5 Les résultats

Dans cette partie nous allons présenter des résultats permettant de valider la technique utilisé pour modéliser une scène 3D avec des ombres douces et des réflexions de l'environnement basé sur le GPU.

III.5.1 La scène finale

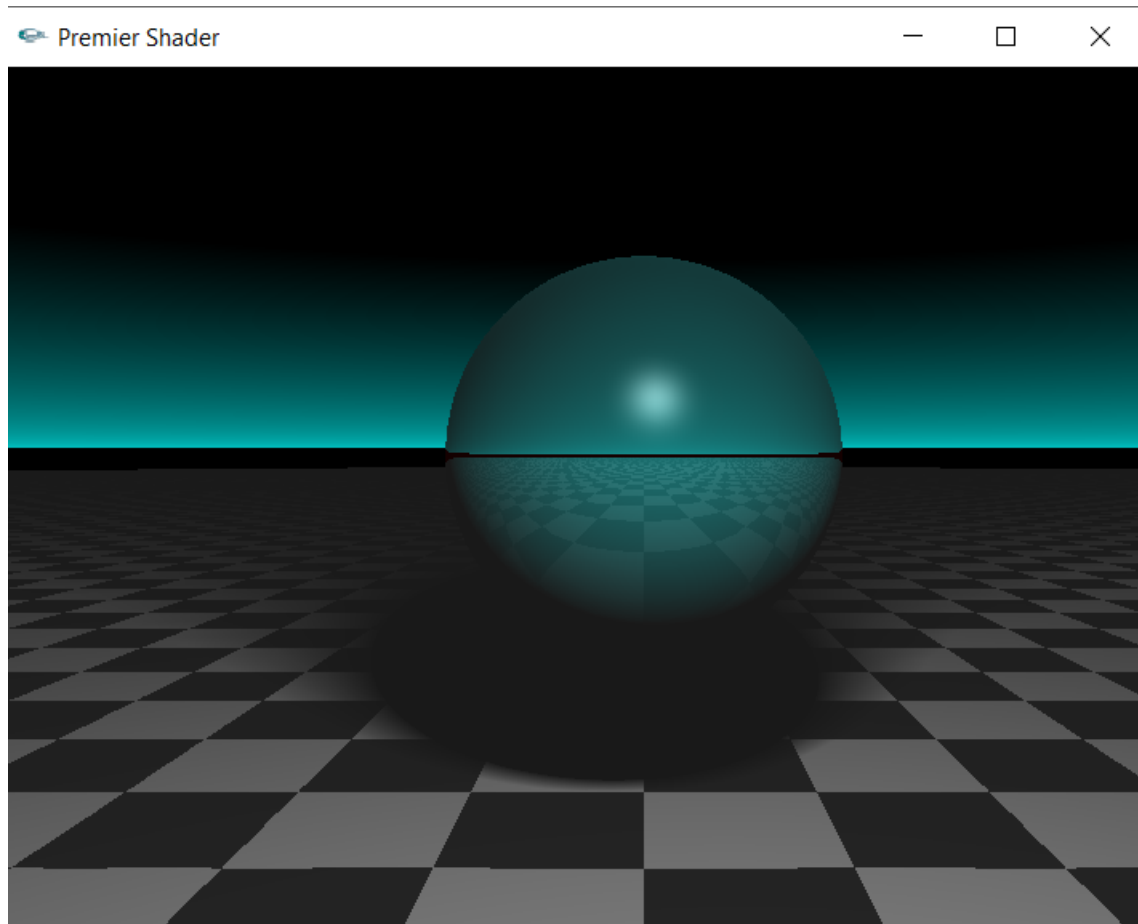


FIGURE III.9 – Le résultat final de shader

III.5.2 Des résultats sans l'apparition de damier sur le sphère

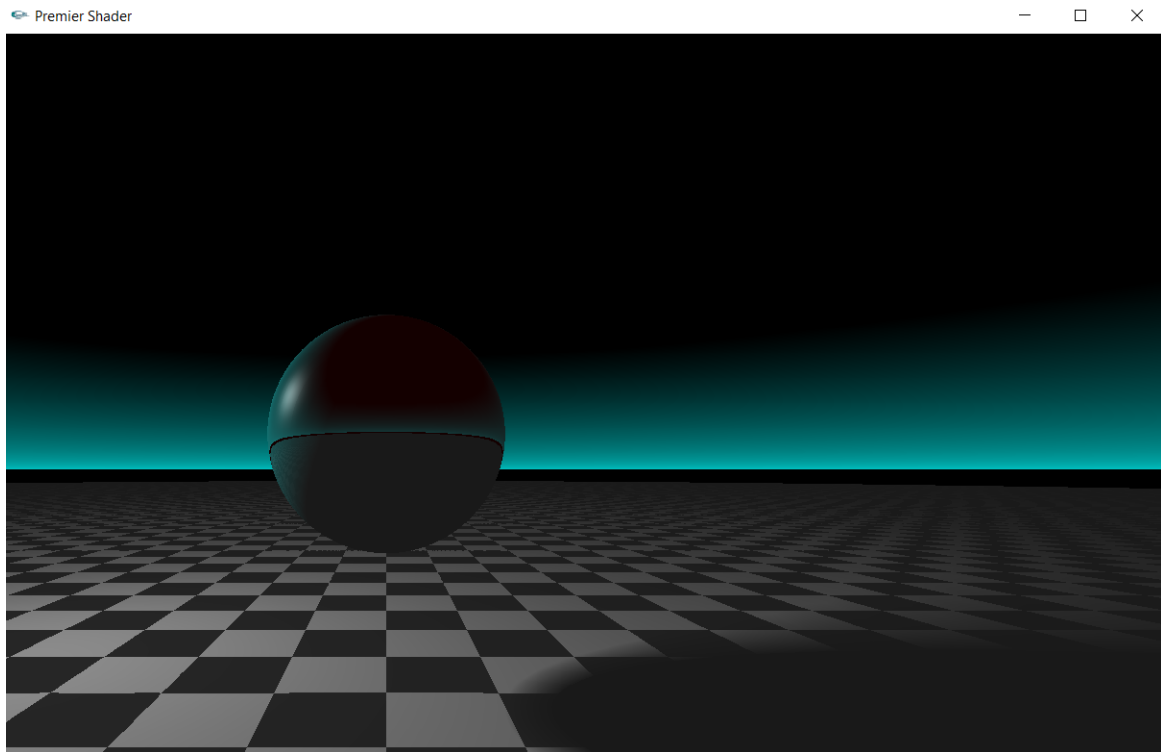


FIGURE III.10 – Des résultats sans l'apparition de damier sur le sphère

III.5.3 Des résultats avec l'apparition de damier sur le sphère

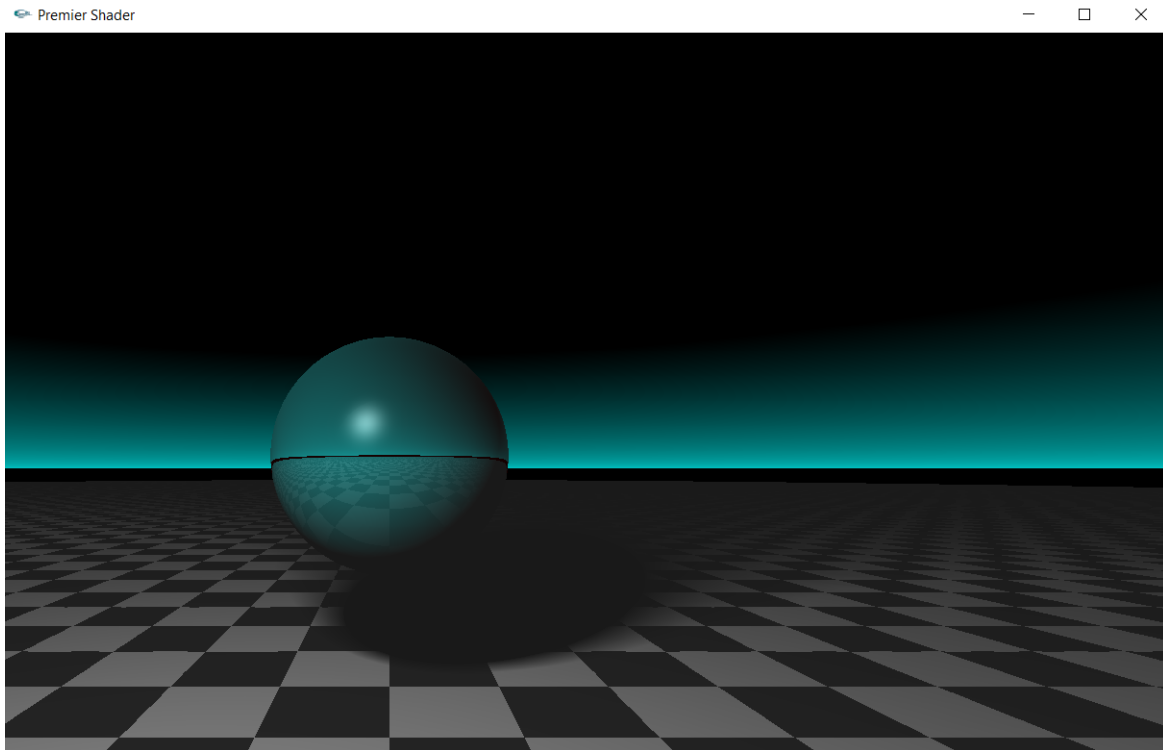


FIGURE III.11 – Des résultats avec l'apparition de damier sur le sphère

III.5.4 Des résultats avec l'images par seconde

Le rendu de l'image est plus que acceptable comme indiqué dans la scène (plus de 200 images par seconde).

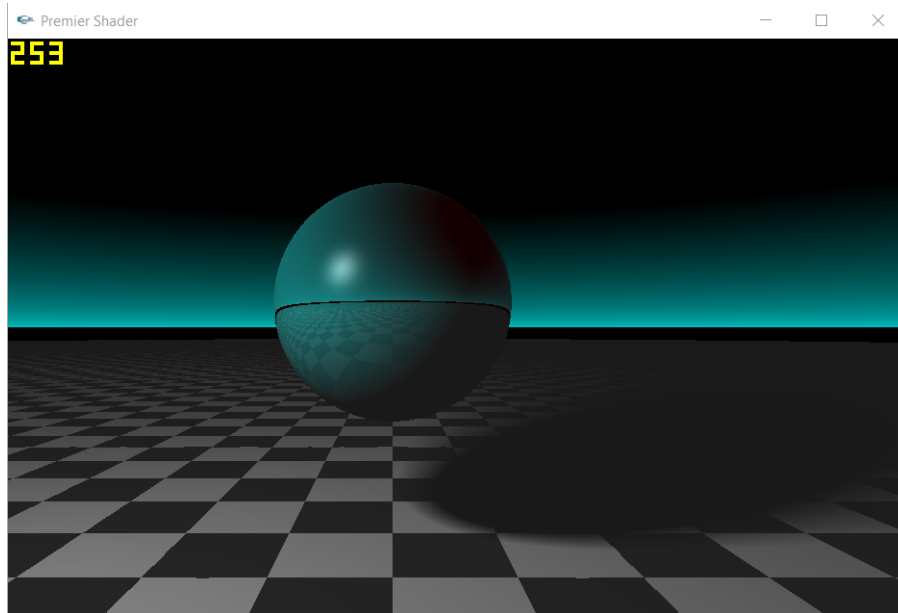


FIGURE III.12 – fps

III.6 Analyse des résultats

Les résultats obtenus sont globalement acceptables et très bons. On voit que les reflets sont tous nets et l'ombrage est excellent à l'aide la technique de ray marching.

Le résultat obtenu est réaliste grâce à la projection de rayons dans la scène, la rotation de sphère avec précision dans l'éclairage et les effets d'ombre dans le damier ont rendu la scène visuellement attrayante.

III.7 Perspectives

Les techniques de ray marching basées sur GPU sont efficaces, mais il est toujours possible d'envisager des optimisations supplémentaires pour accélérer le rendu.

- Cela peut inclure des stratégies pour réduire le nombre de rayons émis, des améliorations des algorithmes de marche ou des techniques d'approximation pour des calculs fastidieux.
- Les techniques de gestion de la complexité des scènes et l'implémentation des structures de données appropriées pour accélérer les intersections d'objets radiaux, ou implémenter un mécanisme d'exclusion pour éviter de traiter des objets invisibles.
- Développement des modèles plus précis pour la génération d'ombre douce et intégration des techniques d'éclairage globales.
- Développement des modèles de matériaux plus avancés et mettre en œuvre des algorithmes de lancer de rayons optimisés pour gérer ces effets.
- Les techniques de marche de rayons basées sur GPU peuvent être étendues pour inclure d'autres effets

visuels tels que l'occlusion ambiante, le flou de mouvement et les effets de particules. Les travaux futurs pourraient se concentrer sur l'intégration de ces effets supplémentaires pour créer des scènes encore plus réalistes et immersives.

- Nous optimisons notre technique de marche de rayons basée sur GPU pour fonctionner efficacement sur ces plates-formes tout en offrant une qualité visuelle satisfaisante.

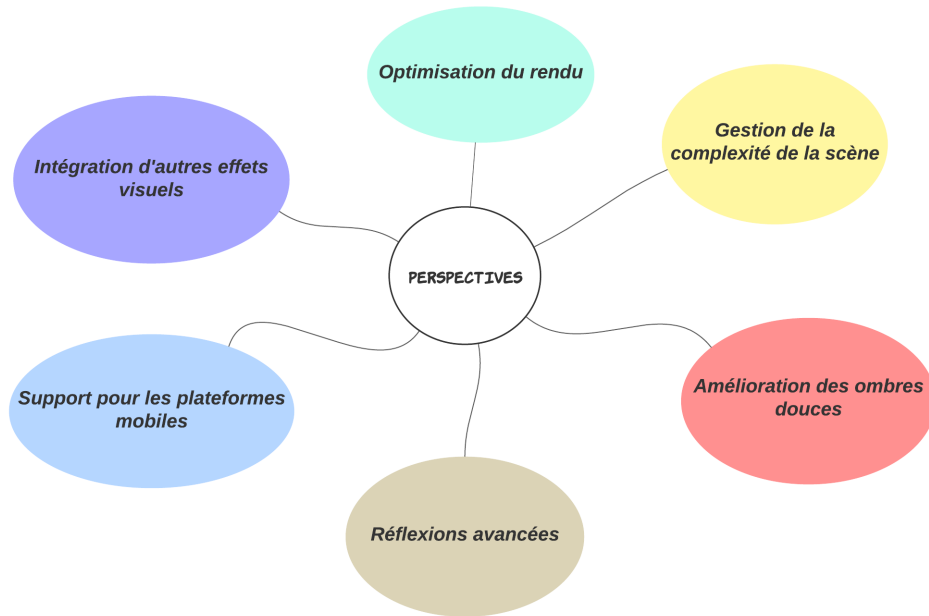


FIGURE III.13 – Travaux futurs à travers lesquels l'amélioration et le développement peuvent être réalisés

Conclusion générale

La modélisation de scène 3D basée sur GPU offre des résultats visuellement incroyablement réalistes. Cette approche permet la simulation d'ombres douces réalistes en tenant compte des interactions complexes entre les objets et la lumière dans la scène.

Le ray marching est une technique de rendu basée sur le lancer de rayons qui calcule efficacement les intersections entre les rayons et la scène 3D. L'utilisation du GPU pour effectuer ces calculs tire parti de la puissance de traitement parallèle de ce composant, ce qui accélère considérablement le processus de rendu.

Les ombres douces sont obtenues à l'aide de plusieurs méthodes d'échantillonnage lors du calcul des ombres pour un point donné. Cela prend en compte la diffusion de la lumière par les objets et crée des ombres plus réalistes avec des transitions douces.

Les réflexions sont également activées par la marche des rayons. En traçant les rayons réfléchis à partir de surfaces réfléchissantes, vous pouvez calculer l'interaction de la lumière avec ces surfaces et obtenir des réflexions précises.

L'utilisation de GPU pour les techniques de marche de rayons améliore les performances et facilite la modélisation de scènes 3D avec des ombres douces et des réflexions en termes de temps de rendu.

La technique de marche de rayons basée sur GPU offre une approche puissante et efficace pour modéliser des scènes 3D avec des ombres douces et des réflexions réalistes. Cela produit des résultats visuellement époustouffants et ouvre de nouvelles possibilités pour les applications de rendu 3D avancées.

Bibliographie

- [1] Modélisation d'illumination globale avec la radiosit , 2022. URL : <https://help.autodesk.com/view/3DSMAX/2024/FRA/?guid=GUID-C5A3C77B-794B-4444-9783-7F2EA11C16BD>.
- [2] D finition de "ombrage". Centre National de Ressources Textuelles et Lexicales, 2023. URL : <https://www.cnrtl.fr/definition/ombrage>.
- [3] Alloprof. Les ombres et les p nombres. Alloprof, Accessed on 2023. URL : <https://www.alloprof.qc.ca/fr/eleves/bv/physique/les-ombres-et-les-penombres-p1065>.
- [4] Johan Arneb ck and Andreas Isberg. Comparing a clipmap to a sparse voxel octree for large scale terrain rendering. *Computer Graphics Forum*, 32(1) :41–50, 2013. URL : <https://www.semanticscholar.org/paper/Comparing-a-Clipmap-to-a-Sparse-Voxel-Octree-for-Arneb%3%A4ck/53a3f2a5f629745643a42658aebd9519e0fbf290/figure/0>.
- [5] Author(s). Ombres douces avec source de lumi re surfacique. *Journal Name*, Volume(Issue) :Page Range, Publication Year.
- [6] Hamza BELAICHE. *illumination r ealiste de terrains en temps r eel*. PhD thesis, UNIVERSITE DE MOHAMED KHIDER BISKRA, 2010.
- [7] David Blythe. Dual-paraboloid environment mapping. Webpage, 1999. URL : <https://www.inf.pucrs.br/flash/tcg/aulas/advanced99/notes/node184.html>.
- [8] Hossein Bonakdari. *Mod lisation des  coulements en collecteurs d'assainissement : application   la conception de points de mesures*. PhD thesis, Universit  de Caen Normandie, 2006.
- [9] Tyler Bovenzi. Raymarch. <https://www.tylerbovenzi.com/RayMarch/>, 2022.
- [10] Pascal B nard. Mod lisation g om trique (des surfaces), 2016.
- [11] Universit  de Californie   San Diego. Cse 167 lecture 13. PDF File, 2018. URL : <https://cseweb.ucsd.edu/classes/wi18/cse167-a/lec13.pdf>.
- [12] R ben Andr  Campos de Carvalho. *Combining the Ray Marching and Rasterization Rendering Models to Provide Real-Time High-Fidelity Graphics*. PhD thesis, Universidade do Porto (Portugal), 2020.
- [13] Geek Culture. An introduction to raymarching : Photorealistic rendering on gpu, 2021. URL : <https://medium.com/geekculture/an-introduction-to-raymarching-photorealistic-rendering-on-gpu-970fdd02e31d>.

- [14] Christopher J Gibson. Point-based color bleeding with volumes. 2011.
- [15] Olivier Godin. Imn528 - chapitre 01. <https://info.usherbrooke.ca/ogodin/enseignement/imn528/chapitres/imn528-chap01.pdf>, 2014.
- [16] Avelina Hadji-Kyriacou and Ognjen Arandjelović. Raymarching distance fields with cuda. *Electronics*, 10(22) :2730, 2021. URL :<https://www.mdpi.com/2079-9292/10/22/2730>.
- [17] Avelina Hadji-Kyriacou and Ognjen Arandjelović. Raymarching distance fields with cuda. In *Proceedings of the International Conference on Computational Science (ICCS)*, pages 217–228, 2021.
- [18] Kyle Hayward. Dual-paraboloid reflections. Blog Post, 2008. URL :<http://graphicsrunner.blogspot.com/2008/07/dual-paraboloid-reflections.html>.
- [19] Hervé Laporte. *Etude logicielle et matérielle d'un système de visualisation temps-réel basé sur la quadratique*. PhD thesis, Lille 1, 1996.
- [20] Guillaume Lavoué. Illumination models. *University of Montreal - Computer Graphics*, 2009. URL :http://www.iro.umontreal.ca/~dift3730/notes/05_illumination.pdf.
- [21] F. Le Grand. Introduction aux rayons : Les rayons lumineux et leur utilisation en optique géométrique. <https://www.f-legrand.fr/scidoc/docmml/graphie/rayons/intro-rayons/intro-rayons.html>.
- [22] Maxime Lhuillier. *Modélisation pour la synthèse d'images à partir d'images*. PhD thesis, Grenoble INPG, 2000.
- [23] HACHAICHI MARIA. Optimisation du rendu des ombres douces en temps réel.
- [24] D.P. Mitchell. A survey of ray casting acceleration techniques. *ACM Transactions on Graphics (TOG)*, 2018.
- [25] Yoshihiro Mizutani and Kurt Reindel. Environment mapping algorithms. Reindel Software. URL :<https://www.reindelsoftware.com/Documents/Mapping/Mapping.html>.
- [26] nicolas holzschuch. Presentation slides. URL :<https://slideplayer.fr/slide/1639050/>.
- [27] Y Pierre. *Chatelier" Une approche de la radiosité par voxels, application à la synthèse d'images*. PhD thesis, Thèse de doctorat, l'Université d'Auvergne Spécialité : Informatique, 4 décembre, 2006.
- [28] Pointers Gone Wild. Optimizing ray marching through partial evaluation, 2016.
- [29] Michaël Rossin. Modélisation par surfaces implicites, 2016.
- [30] Han-Wei Shen. Environment mapping slides. PDF File. URL :https://web.cse.ohio-state.edu/~shen.94/781/Site/Slides_files/env.pdf.
- [31] stevewhims,v-kents,DCtheGeek,msatranjr. Spherical environment mapping. <https://learn.microsoft.com/en-us/windows/win32/direct3d9/spherical-environment-mapping>, 2021.
- [32] Thread. Ray marching distance fields in real-time on the cpu, 2011. URL :<https://board.flatassembler.net/topic.php?p=179865>.

- [33] Lukasz Jaroslaw Tomczak. Gpu ray marching of distance fields. *Technical University of Denmark*, 8, 2012.
- [34] Turais. How to load hdri as a cubemap in opengl. <https://www.turais.de/how-to-load-hdri-as-a-cubemap-in-opengl/>, 2021.
- [35] Michael Walczyk. Ray marching, 2021. URL :<https://michaelwalczyk.com/blog-ray-marching.html>.
- [36] Kevin Watters and Fernando Ramallo. Raymarching toolkit for unity, 2018. URL :<https://kev.town/raymarching-toolkit/gallery/>.
- [37] Wikipedia contributors. Reflection mapping. https://en.wikipedia.org/wiki/Reflection_mapping, 2006.
- [38] Jamie Wong. Ray marching and signed distance functions, 2016. URL :<https://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/>.