

IFT 3150 - Cours Projet

Samuel Fournier

Université de Montreal

12 janvier 2025

Introduction

Description du projet

Ray Tracing

Principe de base du Ray Tracing

Algorithme du Ray Tracing

Primitives supportées

Images générées

Volume Participatif

Explication du Volume Participatif

Implémentation

Test d'intersection avec le volume

Origine dans le volume

Direction nulle

Parcours du volume

Algorithme 1 : Digital Differential Analyzer (DDA)

Algorithme 2 : Ray Marching

Comparaison des deux algorithmes

Atténuation de la lumière

- Principe physique de l'atténuation

- Fonction de phase

 - Fonctions isotropes

 - Fonctions anisotropes

- Implémentation de l'atténuation

- Interpolation tri-linéaire

Shading

- Ancienne implémentation

- Nouvelle implémentation

- Améliorations possibles à la fonction de shading

Conclusion

- Résumé des fonctionnalités implémentées

- Améliorations possibles

 - Optimisation

 - Nouvelles fonctionnalités

- Difficultés rencontrées

Introduction

Description du projet

- ▶ Implémentation d'un volume participatif dans un algorithme de traçage de rayons.

Ray Tracing

Principe de base du Ray Tracing

- ▶ Technique de rendu d'images en 3D.
- ▶ Basée sur le principe de lancer des rayons.
- ▶ Tente de répliquer le tracer de la lumière dans notre monde.
- ▶ Principe d'Helmholtz.

Ray Tracing

Algorithme du Ray Tracing

L'algorithme du Ray Tracing est composé de 3 grandes étapes.

Pour chaque pixel de l'image :

- ▶ Tracer un rayon dans la scène.
 - ▶ Calculer l'origine du rayon.
 - ▶ Calculer la direction du rayon.
 - ▶ Projeter le rayon dans la scène 3D.
- ▶ Tester les intersections avec les objets de la scène.
- ▶ Calculer la couleur des pixels en fonction des intersections.
 - ▶ Calculer la réflexion et/ou la réfraction s'il y a lieu.
 - ▶ Calculer la visibilité (ombre) de chaque lumière (modèle de Blinn-Phong).

Ray Tracing

Primitives supportées

Primitives supportées par le ray tracer du cours IFT 3355 Infographie :

- ▶ Sphère
- ▶ Triangle (Pour les maillages)
- ▶ Cylindre
- ▶ Quad (Plan fini)

Ray Tracing

Images générées

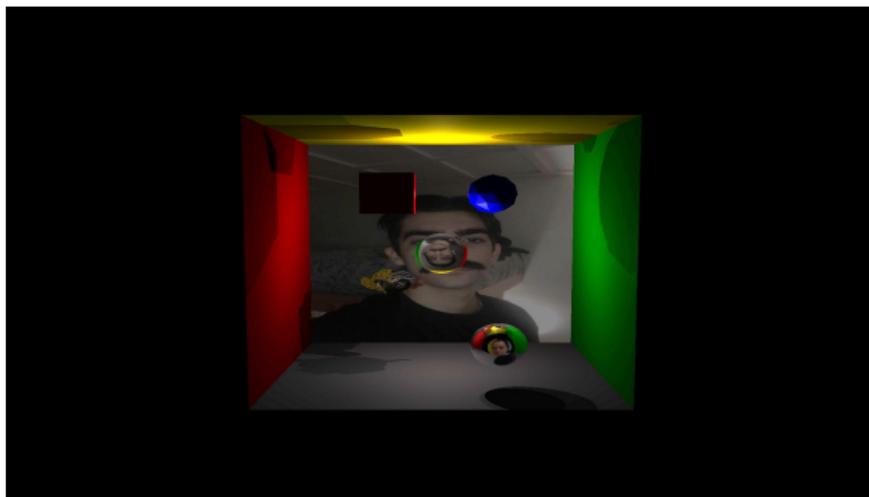


Figure – Image générée par le ray tracer.

Temps de calcul : 462.69s (7.7 minutes)

Volume Participatif

Explication du Volume Participatif

- ▶ Volume 3D.
- ▶ Permet de simuler des effets de fumée, de nuages, de brouillard, etc.
- ▶ Homogène vs. Hétérogène.
- ▶ La lumière interagit avec les particules du volume.

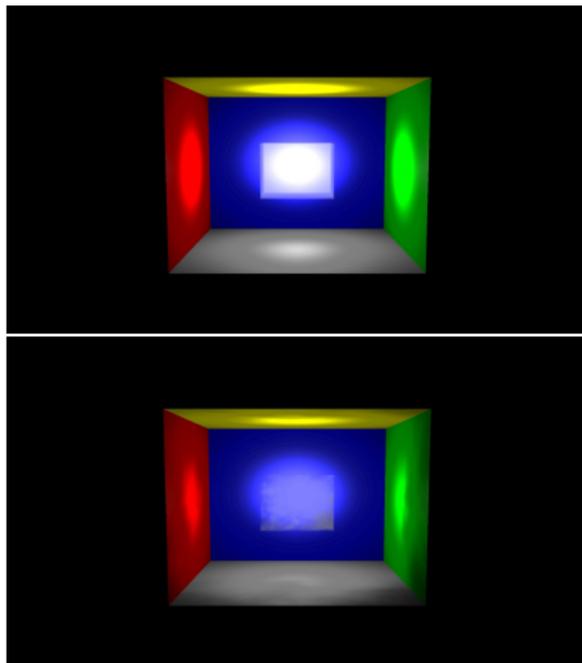


Figure – Homogène vs. Hétérogène

Volume Participatif

Implémentation

- ▶ Ajout d'une nouvelle primitive dans le ray tracer.
- ▶ Ajout d'une nouvelle fonction d'intersection.
- ▶ Modification de la fonction de shading.

Volume Participatif

Test d'intersection avec le volume

Le volume est représenté par une boite englobante (un cube).

L'équation de la boite englobante est :

$$x, y, z \in [0, 1]$$

Pour tester l'intersection avec le volume, on teste l'intersection avec la boite englobante.

Volume Participatif

Test d'intersection avec le volume

L'équation d'un rayon est :

$$r(t) = (o_x + td_x, o_y + td_y, o_z + td_z)$$

L'équation de la boîte englobante est :

$$x, y, z \in [0, 1]$$

On teste l'intersection avec la boîte englobante :

$$o_x + td_x \in [0, 1]$$

$$o_y + td_y \in [0, 1]$$

$$o_z + td_z \in [0, 1]$$

Volume Participatif

Test d'intersection avec le volume - Exemple du développement

Par exemple, en x , on a :

$$o_x + td_x = \{0, 1\}$$

$$t_{x0} = \frac{0 - o_x}{d_x}$$

$$t_{x1} = \frac{1 - o_x}{d_x}$$

Volume Participatif

Test d'intersection avec le volume - Exemple du développement

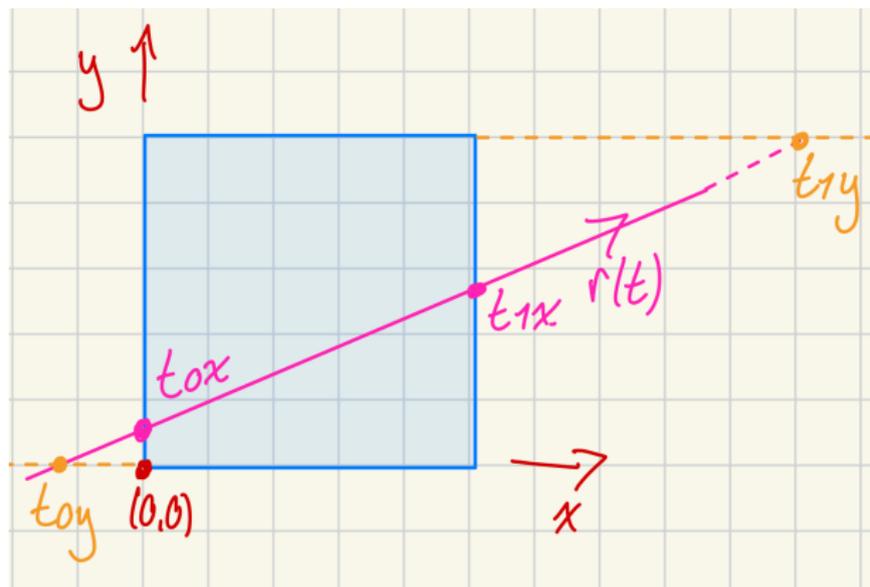


Figure – Visualisation de l'intersection avec la boîte englobante

Volume Participatif

Test d'intersection avec le volume - Origine dans le volume

Cas où l'origine du rayon est dans le volume.

On doit tester une seule intersection avec la boîte englobante.

$$o_x + td_x = \{0, 1\}$$

$$t_{x0} = \frac{0 - o_x}{d_x}$$

$$t_{x1} = \frac{1 - o_x}{d_x}$$

On observe 2 cas distincts :

- ▶ Si $d_x > 0$: $t_{x0} < 0$ et $t_{x1} > 0$.
- ▶ Si $d_x < 0$: $t_{x0} > 0$ et $t_{x1} < 0$.

Volume Participatif

Test d'intersection avec le volume - Direction nulle

Cas où la direction du rayon est nulle.

On se rappelle de la formule d'intersection :

$$o_x + td_x = \{0, 1\}$$

$$t_{x_0} = \frac{0 - o_x}{d_x}$$

$$t_{x_1} = \frac{1 - o_x}{d_x}$$

En C++ (avec c une constante quelconque) :

$$\frac{c}{0} = \pm\infty$$

Parcours du volume

Algorithme 1 : Digital Differential Analyzer (DDA)

Très similaire au test d'intersection avec la boîte englobante.

Trois étapes :

- ▶ Trouver les $tMaxs$ et $tDeltas$ initiaux.
- ▶ Déterminer le plan le plus proche.
- ▶ Avancer vers le plan le plus proche.

Parcours du volume

Algorithme 1 : Digital Differential Analyzer (DDA) - Initialisation

Initialisation des $tMax_x$ et $tDeltas$:

$$tMax_x = \min(t_{x_0}, t_{x_1})$$

$$tMax_y = \min(t_{y_0}, t_{y_1})$$

$$tMax_z = \min(t_{z_0}, t_{z_1})$$

$$tDelta_x = \frac{size_voxel_x}{|d_x|}$$

$$tDelta_y = \frac{size_voxel_y}{|d_y|}$$

$$tDelta_z = \frac{size_voxel_z}{|d_z|}$$

Parcours du volume

Algorithme 1 : Digital Differential Analyzer (DDA) - Déterminer le plan le plus proche

Déterminer le plan le plus proche :

if $tMax_x < tMax_y$ and $tMax_x < tMax_z$ **then**

$tMax_x \leftarrow tMax_x + tDelta_x$

$currentPos \leftarrow \vec{r}(tMax_x)$

else if $tMax_y < tMax_z$ **then**

$tMax_y \leftarrow tMax_y + tDelta_y$

$currentPos \leftarrow \vec{r}(tMax_y)$

else

$tMax_z \leftarrow tMax_z + tDelta_z$

$currentPos \leftarrow \vec{r}(tMax_z)$

end if

Parcours du volume

Algorithme 2 : Ray Marching

Algorithme plus simple que le DDA, mais très similaire.

Deux étapes :

- ▶ Découper le rayon en segments.
- ▶ Avancer de segment en segment avec un pas constant.

Parcours du volume

Algorithme 2 : Ray Marching - Algorithme

```
step ← 0.05 // Valeur arbitraire  
interval ←  $\text{ceil}\left(\frac{t_{\text{Max}} - t_{\text{Min}}}{\text{step}}\right)$   
step ←  $\frac{t_{\text{Max}} - t_{\text{Min}}}{\text{interval}}$   
for n < interval do  
    t ← tMin + (n + jitter()) × step  
    newPos ←  $\vec{r}(t)$   
    evaluateTransmittance(currentPos)  
    evaluateScattering(currentPos)  
    n ← n + 1  
end for
```

Parcours du volume

Comparaison des deux algorithmes

Le DDA est garanti de passer par chaque voxel intersecté par le rayon.

Le DDA échantillonne les voxels une seule fois.

Parcours du volume

Comparaison des deux algorithmes

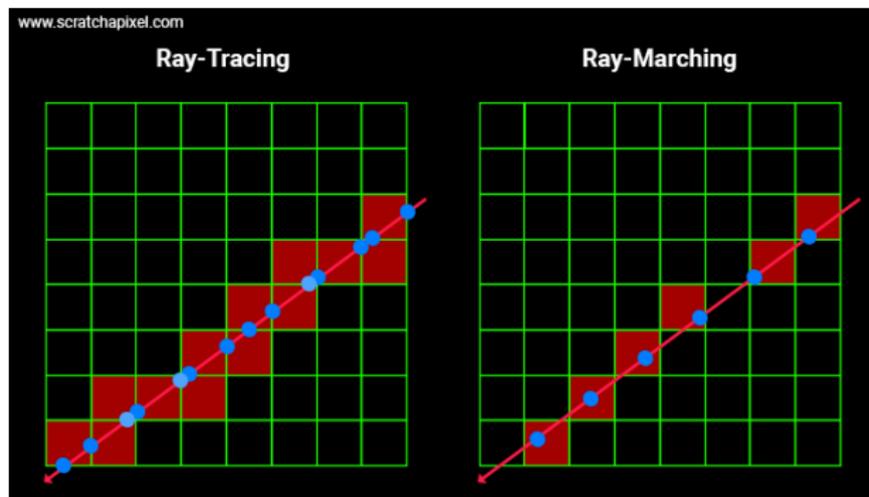


Figure – Comparaison des deux algorithmes

Atténuation de la lumière

Principe physique de l'atténuation

Il y a quatre types d'événement qui peuvent se produire lorsqu'un rayon traverse un volume :

- ▶ Absorption
- ▶ Émission
- ▶ In-Scattering
- ▶ Out-Scattering

Atténuation de la lumière

Principe physique de l'atténuation - Absorption & Émission

L'absorption est le processus par lequel la lumière est absorbée par le volume (σ_a).

L'émission est le processus par lequel le volume émet de la lumière.

Atténuation de la lumière

Principe physique de l'atténuation - In-Scattering & Out-Scattering

L'Out-Scattering est le processus par lequel de la lumière se dirigeant vers la caméra est déviée (σ_s).

L'In-Scattering est le processus par lequel de la lumière ne se dirigeant pas pour la caméra est déviée (σ_s).

Atténuation de la lumière

Fonction de phase

La fonction de phase est une fonction qui détermine la probabilité qu'un photon soit dévié dans une direction donnée.

Il existe deux types de fonction de phase :

- ▶ Isotrope
- ▶ Anisotrope

Atténuation de la lumière

Fonction de phase - Fonction isotrope

La fonction isotrope est une fonction qui dévie les photons de manière uniforme dans toutes les directions.

$$p(\theta) = \frac{1}{4\pi}$$

Atténuation de la lumière

Fonction de phase - Fonction anisotrope

La fonction anisotrope est une fonction qui dévie les photons de manière non-uniforme dans toutes les directions.

$$p(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos \theta)^{\frac{3}{2}}}$$

Si $|g| = 1$, la fonction est nulle.

Si $g = 0$, la fonction est isotrope.

Atténuation de la lumière

Fonction de phase - Autres fonctions anisotropes

- ▶ Schlick

$$p_{\text{schlick}}(\theta) = \frac{1}{4\pi} \frac{1 - k^2}{(1 - k \cos \theta)^2}$$

- ▶ Rayleigh

$$p_{\text{rayleigh}}(\theta) = \frac{3}{16\pi} (1 + \cos^2 \theta)$$

Atténuation de la lumière



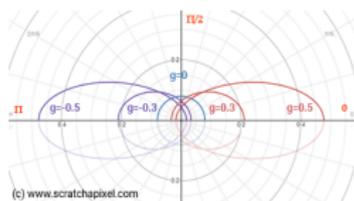
Isotropic



Anisotropic

(c) www.scratchapixel.com

Figure – Différence entre les fonctions isotropes et anisotropes



(c) www.scratchapixel.com

Figure – Différentes valeurs de g pour HG

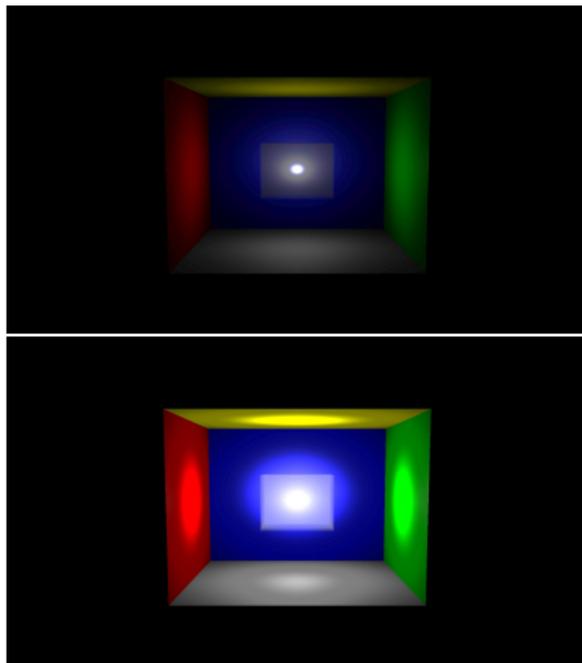


Figure – Comparaison entre $g = -0.65$ et $g = 0.25$.

Atténuation de la lumière

Implémentation de l'atténuation

- ▶ Évaluation de la transmittance avec la loi de Beer-Lambert.
- ▶ Évaluation de la transmittance provenant de la lumière.
- ▶ Évaluation de l'intensité de la lumière.
- ▶ Évaluation de la couleur de la lumière.

Atténuation de la lumière

Loi de Beer-Lambert et Couleur de la lumière

Loi de Beer-Lambert :

$$Attenuation(x) = e^{-step \times (\sigma_a + \sigma_s) \times density}$$

Transmittance :

$$T(x)^* = Attenuation(x)$$

Intensité de la lumière :

$$Li(x) = Le(x) \times Lt(x) \times phasefunc(x)$$

Couleur de la lumière :

$$L(x)^+ = Li(x) \times T(x) \times density \times \sigma_s$$

Atténuation de la lumière

Interpolation tri-linéaire

Utilise les densités des huit voxels qui entourent le point pour interpoler sa densité.

Permet d'avoir une transition plus douce entre les voxels.

Formule de l'interpolation :

$$P(t) = P_0 + (P_1 - P_0)t = P_0(1 - t) + P_1t$$

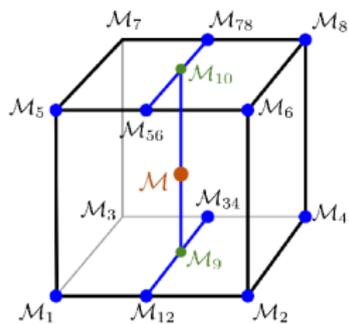


Figure – Interpolation tri-linéaire

Atténuation de la lumière

Interpolation tri-linéaire - Comparaison

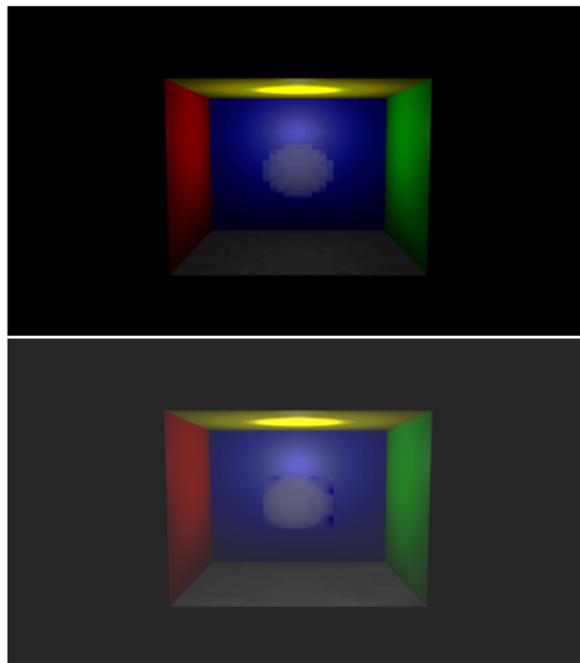


Figure – Comparaison entre aucune interpolation et interpolation tri-linéaire

Shading

Ancienne implémentation

- ▶ Modèle de Blinn-Phong.
- ▶ Implémentation binaire.
 - ▶ Soit l'objet est éclairé, soit il ne l'est pas.
 - ▶ Reçoit 100% de la lumière ou 0%.

Shading

Nouvelle implémentation

- ▶ Modèle de Blinn-Phong.
- ▶ Implémentation continue.
 - ▶ L'objet est éclairé en fonction de la quantité de lumière reçue.
 - ▶ Si un objet bloque la lumière, le point d'intersection n'est pas éclairé.
 - ▶ Si aucun objet ne bloque la lumière, le point d'intersection est éclairé.
 - ▶ Si le volume « bloque » la lumière, le point d'intersection est éclairé en fonction de la transmittance.
 - ▶ Sinon le point d'intersection reçoit 100% de la lumière.
 - ▶ Reçoit un pourcentage de la lumière entre 0% et 100%.

Shading

Améliorations possibles à la fonction de shading

- ▶ Amélioration à la façon dont les objets transparents sont traités.
 - ▶ Lors du shading, on les considère comme des objets opaques.
 - ▶ Produisent des ombres qui n'ont pas de sens.
 - ▶ Utilisé des méthodes comme du *Path Tracing* et *Ray Marching*.

Conclusion

Résumé des fonctionnalités implémentées

- ▶ Implémentation d'un volume participatif.
- ▶ Implémentation de l'atténuation de la lumière.
- ▶ Implémentation de l'interpolation tri-linéaire.
- ▶ Implémentation d'une nouvelle fonction de shading.

Conclusion

Améliorations possibles

- ▶ Optimisation de l'algorithme.
 - ▶ Implémentation de l'algorithme sur GPU (Compute shaders par exemple).
- ▶ Améliorer le traitement des objets transparents.
- ▶ Traiter différents types de volumes.
 - ▶ Nuage, fumée, brouillard, liquide, etc.
- ▶ Implémentation d'un système de particules sur le GPU pour un rendu de volume en temps réel.

Conclusion

Difficultés rencontrées

- ▶ Compréhension des concepts de base de la physique de la lumière.
- ▶ Compréhension des concepts de base de l'atténuation de la lumière.
- ▶ Code en C++ et utilisation des outils de débogage de CLion.

Merci !

