

Precalcul

Nous avons détaillés dans le chapitre précédent les différentes étapes de rendu pour l'intégration d'un objet 3D dans une scène dynamique acquise en temps réel. Certaines des données nécessaires à un tel rendu étant indépendantes de la scène nous nous sommes proposés de les pré-calculer.

Partie 1.1

Éclairage ambiant sous forme de texture

Nous avons vu dans la section ??, page ??, le calcul de l'éclairage ambiant. Dans ce calcul intervenait un terme d'auto-occultation propre à l'objet 3D considéré.

$$\mathcal{P}_V(p) = \frac{1}{\pi} \int_{V(p, \vec{n}(p))} \vec{\omega} \cdot \vec{n} d\vec{\omega} \quad (1.1.0.1)$$

1.1.1

Intégration statistique

Si s'agit donc d'évaluer, en tout point de l'objet, l'intégrale du produit scalaire entre le vecteur d'intégration et la normale à l'objet, pour un vecteur d'intégration parcourant l'espace visible.

On peut alors reformuler le calcul comme suit.

$$\begin{aligned} \mathcal{P}_V(p) &= \frac{1}{\pi} \int_{V(p, \vec{n}(p))} \vec{\omega} \cdot \vec{n} d\vec{\omega} \\ &= \frac{1}{\pi} \int_{\mathcal{H}^2(\vec{n}(p))} \vec{\omega} \cdot \vec{n} \times \text{visible}(p, \vec{\omega}) d\vec{\omega} \\ &= \frac{1}{\pi} \int_{\mathcal{S}} \vec{\omega} \cdot \vec{n} \times \text{HS}(\vec{\omega} \cdot \vec{n}) \times \text{visible}(p, \vec{n}) d\vec{\omega} \end{aligned} \quad (1.1.1.1)$$

Pour évaluer une telle intégrale, on procède par une intégration statistique (Monte Carlo). Pour cela on simule le domaine d'intégration sur l'espace de définition en tirant n points uniformément sur une sphère. Ces points représentent les vecteurs $\vec{\omega}$.

Pour chacun de ces points on s'agit d'évaluer :

$$\vec{\omega} \cdot \vec{n} \times \text{HS}(\vec{\omega} \cdot \vec{n}) \times \text{visible}(p, \vec{n})$$

Le terme $\text{HS}(\vec{\omega} \cdot \vec{n})$ est en fait inutile car tout point pour lequel $\text{visible}(p, \vec{n})$ sera non nul sera assuré d'être dans l'hémisphère décrit par $\text{HS}(\vec{\omega} \cdot \vec{n})$. La fonction à intégrer peut donc se résumer à :

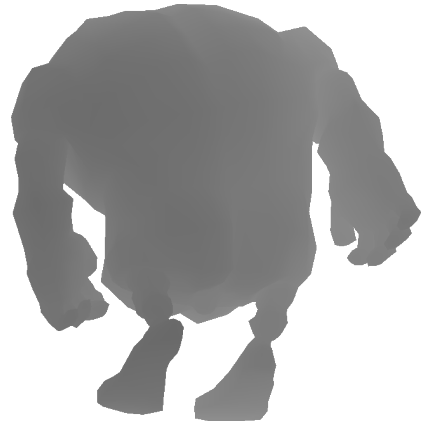
$$\vec{\omega} \cdot \vec{n} \times \text{visible}(p, \vec{n})$$

1.1.2

Évaluation de la visibilité

L'évaluation de la visibilité est faite à l'aide d'une shadowmap. Étant donnée un point de vue (vecteur $\vec{\omega}$ aléatoire) il s'agit d'effectuer une première étape de rendu, avec une projection orthographique pour en conserver que les informations de profondeurs retenues dans le Z-buffer (voir figure 1.1).

FIGURE 1.1 Z-buffer issue de la première étape de rendu (shadowmap)



Ces données de profondeurs permettent ensuite d'évaluer si un texel est visible ou non, simplement en comparant la profondeur du texel considéré et celui retenu dans la texture et qui caractérise le texel le plus proche selon le rayon associé.

La seconde étape de rendu consiste alors à écrire, dans l'espace texture associé à l'objet, les informations de visibilité qui auront été calculées (voir figure 1.2). À cette étape, il est important de s'assurer que la texture est bien écrite, y compris au niveau des jointures entre les différents éléments.

1.1.3

Post traitement

Une fois évalué pour une source, les résultats sont ajoutés à une texture qui somme les contributions des différentes sources. Cette texture contient ainsi le résultat de l'intégration considérée.

Une fois toutes les sources évaluées, il reste quelques étapes de post traitement :

- L'intégration étant normalisée, les valeurs calculées sont théoriquement entre 0.0 et 1.0. Cependant du

FIGURE 1.2 Rendu en espace texture pour une source



fait de la distribution aléatoire des sources utilisées pour l'intégration, il peut arriver qu'en certains points ne présentant pas d'auto-occultation la valeur dépasse 1.0. Les valeurs sont alors seuillées afin de ne pas poser de problème au moment du stockage au format .png.

Les données étant quantifiées entre 0 et 255, un dépassement de la valeur flottante à stoker peut provoquer une traduction en un entier supérieur à 255. Ces nombres étant stockés sous forme de caractère ASCII, un dépassement à 256 ou 257 peut alors provoquer une évaluation modulo 256 soit un stockage sous forme de 0 ou de 1.

- Les parties de l'image ne correspondant pas a des coordonnées de textures valides étant jusque la vides. L'évaluation des niveaux de mipmap pour la texture considéré risque de prendre en compte des éléments qui ne sont par représentatives de la réalité géométrique de l'objet. Afin de limiter ce biais dans l'évaluation des niveaux de mipmap, on remplit les parties vides et non représentative avec la valeur moyenne calculée sur les parties représentatives. Ainsi on limite l'incohérence des données considère en bordure de patch pour les niveaux de mipmap les plus élevés.

A l'issue de ces différents post-traitement, il ne reste qu'a exporter la textures sous forme d'image (voir figure 1.3) qui sera chargée le moment voulu.

Partie 1.2

Décomposition en sphères

La section ??, page ??, traite du rendu d'ombres portées issus de l'objet 3D ajouté à la scène. Cette étape nécessite une représentation de l'objet en question comme union de sphères.

Le pré-calcul étudié ici consiste donc à construire, étant donné un maillage, à calculer un ensemble limité de sphère qui s'approche au mieux de la surface correspondant au maillage.

FIGURE 1.3 Données pré-calculées en espace texture après post-traitement (1000 sources)

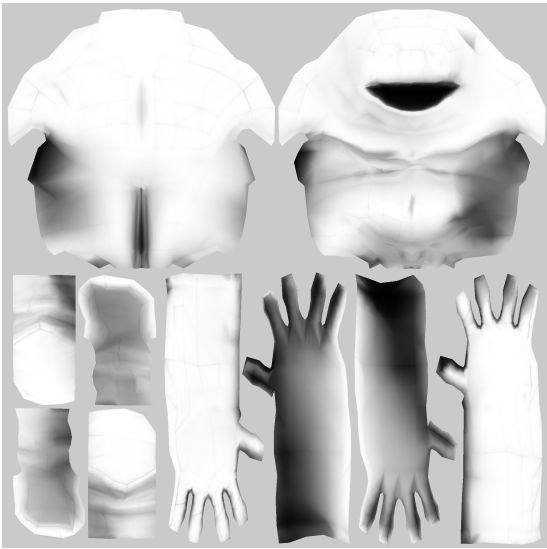
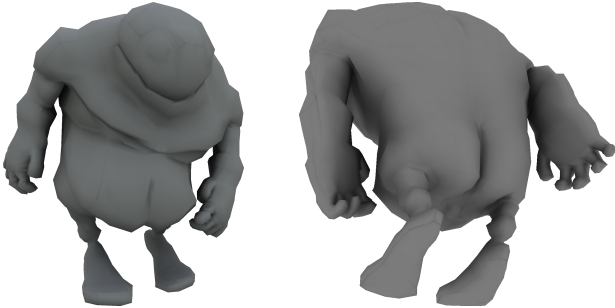


FIGURE 1.4 Utilisation des données d'éclairage ambiant lors du rendu



1.2.1

Échantillonnage du maillage

Afin d'obtenir une distribution uniforme de points à la surface de l'objet, on procède a un échantillonnage par importance¹.

Pour construire un point uniformément a la surface du maillage on sélectionne d'abord une face proportionnellement à sa surface avant de choisir un point uniformément sur la face considéré.

En répétant cette opération un grand nombre de fois on obtient un nuage de points qui représente correctement les grandes faces. Ce sont ces points qui seront utilisés par la suite pour la construction des sphères représentatives.

1.2.2

Appareillement de points et calcul de sphère optimale

La phase d'appareillement de l'algorithme consiste à organiser les points en clusters relatifs aux sphères déjà construites. Il s'agit donc d'établir une métrique qui caractérise la proximité d'un point à une sphère. De la même manière, la phase de calcul de sphère optimale consiste à calculer la sphère qui minimisera la métrique considère précédemment sur le cluster qui lui a été associé.

1. Importance sampling

Algorithme 1.2.1 : Ajustement de sphères à un maillage

Entrées : Maillage m , int n , int s

```

/* Échantillonnage du maillage */
1 Point  $pts[s]$ ;
2 pour  $i \leftarrow [1..s]$  faire
3    $pts[i] \leftarrow \text{EchantionnageUniforme}(m)$ ;

/* Initialisation des sphères */
4 pour  $i \leftarrow [1..n]$  faire
5    $sphs[i] \leftarrow pts[i]$ ;

6 tant que  $sphs$  n'est pas stable faire
7   Point  $clsts[n][ ]$ ;
8   pour  $i \leftarrow [1..n]$  faire
9      $clsts[i] \leftarrow \text{Appareillement}(pts, sphs[i])$ ;
10  pour  $i \leftarrow [1..n]$  faire
11     $sphs[i] \leftarrow \text{SphereOptimale}(clsts[i])$ ;
12 retourner  $sphs$ ;
    
```

La principale difficulté ici est de construire un métrique représentative des critères de qualités voulus, pour laquelle le calcul d'un élément optimal n'est pas trop complexe et qui converge vers un état stable.

Une métrique envisagé est d'évaluer la distance d'un point à la surface de la sphère associé :

$$\begin{aligned}
 f(P, S) &= \|P - S\|^2 - S.r^2 & (1.2.2.1) \\
 &= (P.x - S.x)^2 + (P.y - S.y)^2 + (P.z - S.z)^2 - S.r^2 \\
 &= (P.x^2 + P.y^2 + P.z^2) + (S.x^2 + S.y^2 + S.z^2) \\
 &\quad - 2(P.xS.x + P.yS.y + P.zS.z) - S.r^2 \\
 &= \|P\|^2 + \|S\|^2 - 2(P.xS.x + P.yS.y + P.zS.z) - S.r^2
 \end{aligned}$$

Ce calcul de la sphère optimal associé a un cluster peut ensuite se faire par l'optimisation d'un système matriciel $AX - B$, la matrice A représentant les données en entrée (points échantillonnées), le vecteur X représentant la solution à optimiser et le vecteur B étant l'objectif à atteindre.

L'équation 1.2.2.1 nous indique comment construire A , X et B :

$$A = \begin{pmatrix} -2P_1.x & -2P_1.y & -2P_1.z & 1 \\ -2P_2.x & -2P_2.y & -2P_2.z & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -2P_n.x & -2P_n.y & -2P_n.z & 1 \end{pmatrix} \quad (1.2.2.2a)$$

$$X = \begin{pmatrix} S.x \\ S.y \\ S.z \\ \|S\|^2 - S.r^2 \end{pmatrix} \quad (1.2.2.2b)$$

$$B = \begin{pmatrix} -\|P_1\|^2 \\ -\|P_2\|^2 \\ \vdots \\ -\|P_n\|^2 \end{pmatrix} \quad (1.2.2.2c)$$

Ainsi on a :

$$A.X - B = \begin{pmatrix} f(P_1, S) \\ f(P_2, S) \\ \vdots \\ f(P_n, S) \end{pmatrix} \quad (1.2.2.2d)$$

Cette première considération géométrique n'est cependant pas satisfaisante. La simple proximité entre un point et la surface d'une sphère ne prend pas en compte les différences d'orientation de normale entre une face du maillage et la sphère qui le modélise, lesquels sont des éléments de qualité essentiels dans notre cas. Au delà de ce problème d'orientation, cette métrique fait aussi face au problème des outliers présents sur les modèles non convexes. Si quatre points extrémaux se retrouvent loin de tout, il est possible qu'une sphère viennent les adopter comme point d'attache, englobant alors tout le modèle et cachant la géométrie que l'on voulait représenter.

Par ailleurs la métrique proposé précédemment ne garanti pas la convergence du système.

Finalement la solution adopté, bien que géométriquement pauvre, nous donne des résultats visuellement acceptables tout en garantissant une convergence rapide. L'idée est de paver la surface du maillage à l'aide de sphères de rayon uniforme, ce qui correspond à la minimisation de la distance entre les points et le centre des sphère qui leur sont associés.

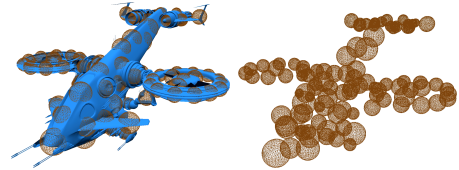
La distance considérée

$$f(P, S) = \|P - S\| \quad (1.2.2.3)$$

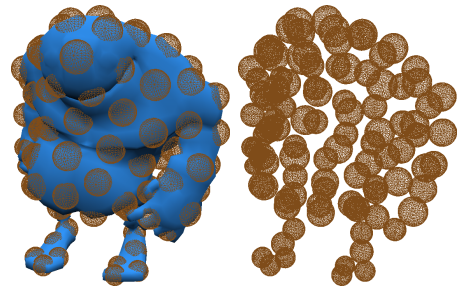
s'optimisant trivialement sur un cluster C par

$$\bar{S} = \left\{ \left(\bar{p}, \overline{\|p - s\|} \right) \mid p \in C \right\} \quad (1.2.2.4)$$

FIGURE 1.5 Décomposition en sphères d'objets 3D (100000 échantillons, 100 sphères)



(a) MRX22 (157781 sommets, 287776 triangles)



(b) Bigguy (1754 sommets, 2900 triangles)

Bibliographie
