Lancement, méthodes et algorithmes

Moment Based Rendering

Baptiste Delos, Mehdi Djemai, Alban Odot,
Pierre Mézières et Jean-Baptiste Sarazin

Encadrant : Mathias Paulin

Master 2 - Informatique Graphique et Analyse d'Images

Application

Figure 1: Exemple d'ombrage

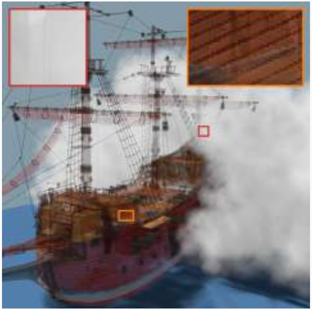


Figure 2: Exemple de transparence

Sommaire

Lancement

Plan de développement

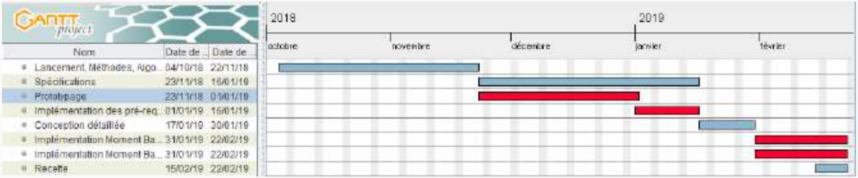


Figure 3: Diagramme de Gantt

Conception

Développement

Organisation du projet

- **Communication interne**: Discord, Google drive
- **Communication externe**: Mail, Réunion bimensuelle si nécessaire
- **Gestion des versions**: Git
- **Modèle de développement**: Itérations de 2 semaines
- **Débogage**: Critère Visuel, Valgrind ...
- **Test**: Test spécifique sur le code jugé sensible

Méthodes

Attentes et vision globale des articles

Travail à réaliser:

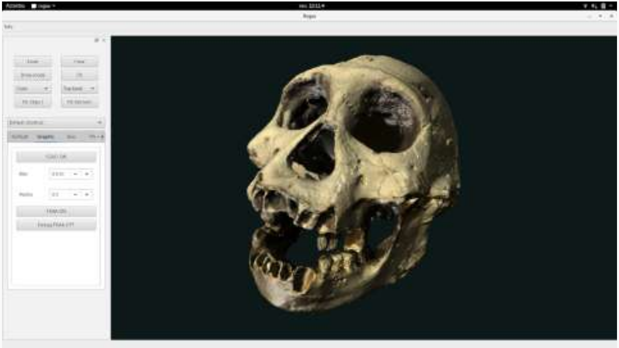
- Algorithmes d'ombrage
- Transparence avec indépendance à l'ordre des fragments

$\rightarrow$ Création et modification de *renderer* dans un moteur 3D temps réel.

But: Comparer les algorithmes de bases avec les algorithmes présentés dans les papiers.

Point de départ

Moteur 3D: Rogue (Render Object Graphic Useless Engine)



Organisation globale:

1. Assurer compilation Mac
2. Implantation *renderer* pour l’ombre et la transparence de base
3. Implantation des algorithmes présentés dans les papiers

Profondeur non linéaire → ~~Filtrage linéaire~~

Idée:
Représenter la profondeur avec des données linéaires.

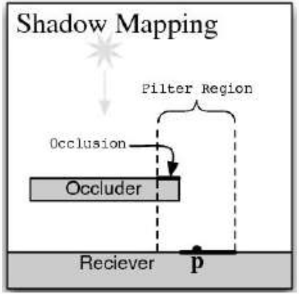


Figure 4: Représentation d'une profondeur non linéaire

Comparaisons

Techniques de référence

- Ombrage $\rightarrow$ Percentage Closer Filtering
- Transparence $\rightarrow$ *Depth Peeling*

Mise en place de la comparaison

- **Métrique visuelle**: Différence d'images, distance entre couleurs
- **Temps d'exécution**: Mesure de génération d'une image

Algorithmes

Moment Shadow Mapping - Généralités

b est la donnée stockée dans la shadow map

1. Classique:
$$b : [0,1] \to \mathbb{R}.$$
2. Général:
$$b : [0,1] \to \mathbb{R}^m.$$
3. Nous:
$$b : [0,1] \to \mathbb{R}^4.$$

Moment Shadow Mapping - Principe

Echantillonner une shadow map via une variable aléatoire sur $\mathbb{R}^2$.

1. Soit $t : \mathbb{R}^2 \to \mathbb{R}^2$ une variable aléatoire.
2. Soit $s : \mathbb{R}^2 \to [0, 1]$ une shadow map.

$\mathsf{z} := s \circ t \in [0, 1]$
$Z := P_{\mathsf{z}} : \mathcal{B}([0, 1]) \to [0, 1]$ est la distribution de profondeur par le noyau t sur la shadow map s.

Moment Shadow Mapping - Linéarité

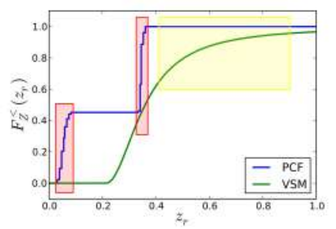
$\mathcal{B}$ étant une σ-algèbre sa structure d'espace vectorielle nous donne

$$(\omega_1 P_1 + \omega_2 P_2)(\omega) = \omega_1 P_1(\omega) + \omega_2 P_2(\omega)$$

Combinaison linéaire de profondeurs $\rightarrow$ combinaison linéaire des ombres
Coefficients linéairement filtrables $\rightarrow$ anti-aliasing sur la shadow map

Moment Shadow Mapping - Approximation de la PCF

- **Objectif** : Approximer la fonction de visibilité
- Reconstruction linéaire impossible aux bords des objets.
- **Minorer** la visibilité : *Light leaking*
- **Majorer** la visibilité : Auto-occlusion



On choisit de minorer la visibilité car les artefacts sont moins gênants.

- *VSM* : Méthode non-linéaire qui utilise $\mathbb{E}(z)$ et $\mathbb{E}(z^2)$
- *Four Moments Shadow Mapping* : Offre une meilleure approximation

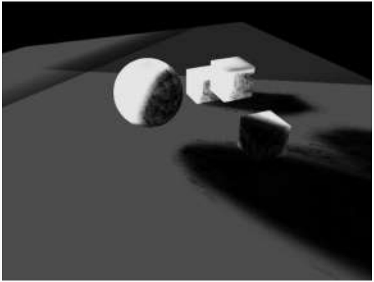


Figure 5: Artefacts générés par le VSM

Moment Shadow Mapping - Espace de recherche

Espace de recherche

- Soit $I \subset \mathbb{R}$.
- Soit $\mathbf{b} : I \to \mathbb{R}^m$.
- Soit $Z \in \mathfrak{P}(I)$. On pose $b := \mathbb{E}_Z(\mathbf{b})$

$$\mathfrak{S}_b := \{S \in \mathfrak{P}(I) | \mathbb{E}_S(\mathbf{b}) = b\}$$

Trouver le minorant optimal pour $F_S(z_r) = P([-\infty, z_r])$ définie par

$$\rho_I(b, z_r) := \underset{S \in \mathfrak{S}_b(I)}{inf} F_S(z_r)$$

Moment Shadow Mapping - Solution

Entrées: $T := \{z_1, ..., z_n\} \subset [0,1], \mathbf{b} : [0,1] \to \mathbb{R}^m, z_r \in [0,1]$
Sortie: $Z \in \mathfrak{S}_b \cap \mathfrak{P}(T)$ avec $F_Z(z_r) = \rho_T(b, z_r)$

---

1. $A := (\mathbf{b}_j(z_i)) \in \mathbb{R}^{m \times n}$
2. $\bar{A} := \begin{pmatrix} 1 \cdots 1 \\ A \end{pmatrix}$
3. $\bar{b} := \begin{pmatrix} 1 \\ b \end{pmatrix}$
4. $p \in \mathbb{R}^n$ avec $p_i := \chi_{-\infty, z_r}(z_i)$
5. **Trouver** ω minimisant $p^T.\omega$ sujet à la contrainte $\bar{A}.\omega = b$ et $\omega_i \geq 0 \forall i \in \{1, ..., n\}$
6. **Retourner** $Z = \sum_{i=1}^{n} \omega_i \delta_{z_i}$ en cas de succès
   Indiquer $\mathfrak{S}_b \cap \mathfrak{P}(T) = \varnothing$ sinon

Moment Shadow Mapping - Hamburger/Hausdorff 4MSM

Hamburger 4MSM :

1. $b := \mathrm{b}_{MSM} := (z, z^2, z^3, z^4)^T$.
2. Défini sur $\mathbb{R}$
3. **Insensible** à la différence Z-Near, Z-Far

Hausdorff 4MSM :

1. $b := \mathrm{b}_{MSM} := (z, z^2, z^3, z^4)^T$.
2. Défini sur $[0, 1]$
3. **Sensible** à la différence Z-Near, Z-Far

Problème de la transparence

Equation de mélange :

- $n \in \mathbb{N}$ fragments
- L_l Couleur du fragment l
- α_l Opacité du fragment l
- z_l Profondeur du fragment l

$$\sum_{l=0}^{n-1} L_l \cdot \alpha_l \cdot T(z_f)$$

Problème de la transparence

Transmittance :

$$T(z_f) = \prod_{\substack{k=0 \\ z_k < z_l}}^{n-1} (1 - \alpha_k)$$

Taux d'absorption:

$$A(z_f) := -\ln(T(z_f)) = \sum_{\substack{l=0 \\ z_l < z_f}}^{n-1} -\ln(1 - \alpha_l)$$

Composition dépendante à l'ordre des fragments

Objectif des méthodes *Order Independant Transparency :*
Représenter $A(z_f)$ sans tri préalable des fragments

Moment Based Order Independant Transparency

Cumulative :

$$Z := \sum_{l=0}^{n-1} - \ln(1 - \alpha_l) \cdot \delta(z_l)$$

Espérance :

Hamburger 4MSM propose $\mathrm{b}(z) = (1, z, z^2, z^3, z^4)$

$$b := \mathbb{E}_z(\mathrm{b}) := \sum_{l=0}^{n-1} - \ln(1 - \alpha_l) \cdot \mathrm{b}(z_l)$$

b est l'absorbance du fragment à la profondeur z indépendemment de son ordre.

Moment Based Order Independant Transparency

Couleur :

$$\exp(-b_0) \cdot L_n + \Gamma \cdot \sum_{l=0}^{n-1} L_l \cdot \alpha_l \cdot T(z_f, b, \beta)$$

avec

$$\Gamma = \frac{1 - \exp(-b_0)}{\sum_{l=0}^{n-1} \alpha_l \cdot T(z_f, b, \beta)}$$

un facteur de normalisation

Minimisation des erreurs a priori

Depth Warping :

$$z := \frac{ln(z_v) - \ln(z_{min})}{\ln(z_{max}) - \ln(z_{min})} \cdot 2 - 1$$

avec $z_v \in \mathbb{R}$ profondeur dans l'espace de vue

Application d'une transformation affine Θ_m^* au vecteur $\mathbf{b}(z)$ (moments exponentiels)

Questions ?