

Photon mapping

Babahenini Mohamed Chaouki

Motivations

- L'objectif du placage de photons est de fournir un algorithme efficace pour faire le **rendu d'images complexes avec de l'illumination globale**.
- Le placage de photons propose de combler certaines lacunes des méthodes de radiosit  :
 - Rendu de **surfaces sp culaires** tr s difficile ;
 - **Temps de calcul astronomique** d    la grande complexit  algorithmique.
- On a vu une m thode permettant   la fois de faire le rendu des r flexions diffuses et sp culaires : le **trac  de chemins**.
- Cette poss de quelques caract ristiques digne d'int r t :
 - Possibilit  de simuler les effets d'illumination locale et globale ;
 - G om trie arbitraire sans maillage ;
 - Possibilit  d'obtenir des r sultats r alistes.

Motivations

- Le problème principal du tracé de chemins est **la variance dans les résultats**, qui est perçue sous forme de bruit.
- Pour éliminer ce bruit, on devait **lancer davantage de rayons aléatoires**, ce qui devenait rapidement très coûteux.
- Le placage de photons se veut **une alternative efficace au tracé de chemins**.
- Il propose de conserver les avantages de celui-ci, tout en **produisant des résultats exempts de bruit** de haute fréquence.

L'idée de base

- Une idée de base du placage de photons et de **séparer la représentation de l'illumination de celle de la géométrie** de la scène.
- Pour ce faire, les données d'illumination seront stockées dans leur propre structure de données : **la carte de photons**. La construction de cette carte est une étape importante du placage de photons.
- Le placage de photons est une technique qui s'effectue en deux phases :
 - **Traçage de photons** : Construction de la carte de photons en lançant ceux-ci dans le modèle géométrique à partir des sources de lumière.
 - **Rendu de la scène** : À partir de la carte de photons, faire un rendu efficace de l'environnement.

Définitions et principes

Un algorithme à deux passes pratique et robuste :

- Trace les chemins d'illumination à la fois depuis les sources et depuis le point de vue.
- Met en cache et réutilise les valeurs d'éclairage dans une scène pour plus d'efficacité.
- Le flux de la source de lumière est réparti entre les trajets lumineux sortants (trajectoires de photons) et transporté vers la scène.
- La densité des photons est capturée par les chemins de caméra.
- Le Photon mapping est biaisé

Principe du photons mapping

- **Première passe:**

- Les photons sont tracés à partir des sources de lumière dans la scène
- Ils transportent des informations de flux
- Ils sont mis en cache dans une structure de données, appelée carte de photons

- **Deuxième passe:**

- une image est rendue en utilisant les informations stockées dans la carte de photons

Avantages du tracé de particules

- Ne nécessite pas des surfaces Lambert (Lambertiennes ou parfaitement diffuses)
- Peut interagir avec les milieux participants
 - Les milieux participants (espace vide ou milieu perméable hétérogène) peuvent atténuer les particules
 - Les milieux participants peuvent stocker des photons, donc ils peuvent être allumés!
- Peut modéliser statistiquement le comportement dépendant de la longueur d'onde de la lumière et des matériaux
- Peut modéliser des effets d'inter-réflexion spéculaire (par exemple caustiques)

Carte de photons

- stocke les effets de la surface des rayons lumineux
 - la **position** de l'intersection ;
 - la **puissance** du photon (flux radiant) ;
 - la **direction incidente** du photon
- Durant le traçage de photons, **les photons sont stockés dans un tableau 1D** qui n'est aucunement lié à la géométrie de la scène.
- Au moment du rendu, l'intensité d'un pixel de l'image est calculée à partir des photons les plus près du point 3D de la scène correspondant au pixel.

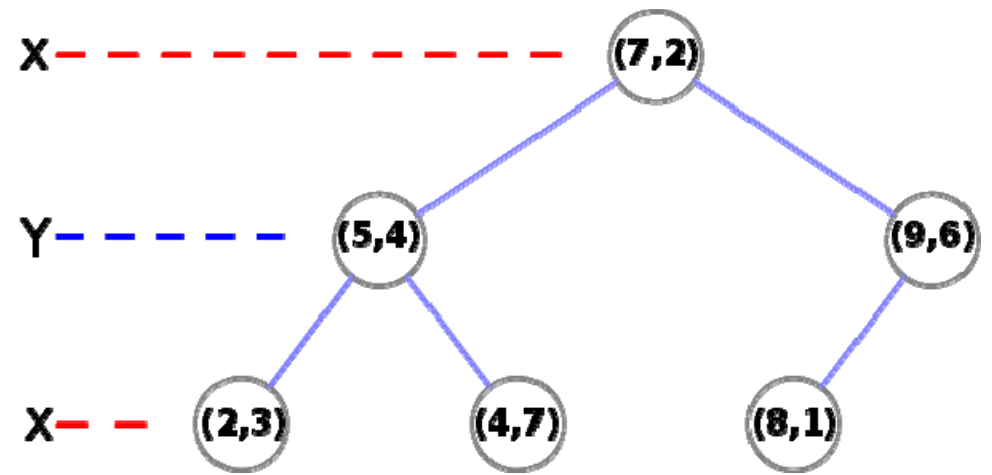
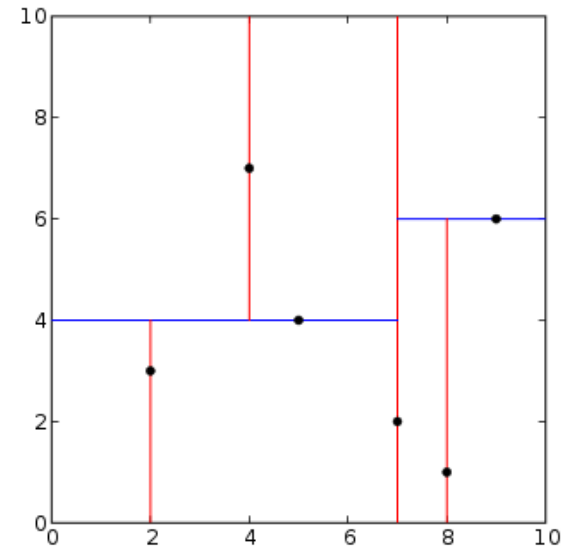
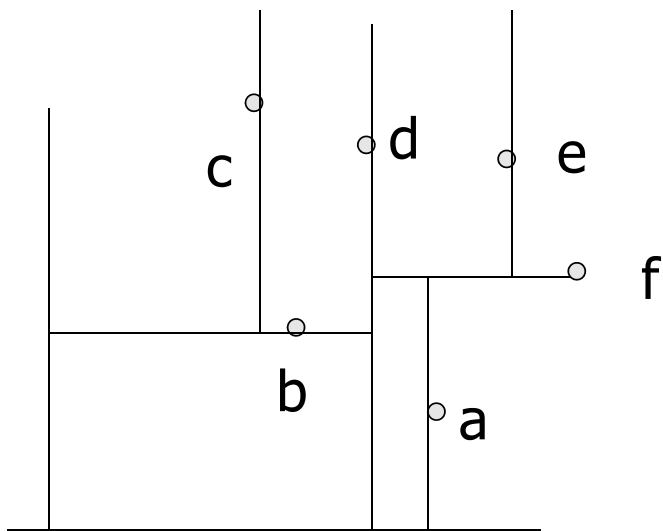
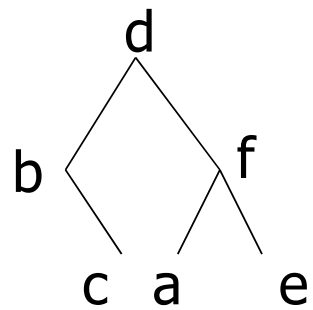
Structure de la carte de photons

- Pour que l'algorithme du placage de photons soit efficace, la structure de données choisie pour stocker les photons de la scène doit **permettre de rapidement trouver les voisins d'un point 3D** parmi un ensemble de points 3D.
- Une structure efficace pour contenir une distribution non uniforme de points est le kd-tree (avec $k = 3$ pour une carte de photons).
- **Chaque noeud de l'arbre contient un photon** et des pointeurs vers les sous-arbres de gauche et de droite.
- Utilise une structure de données 3D distincte:
 - Sépare le stockage des photons du paramétrage de la surface
 - Traite la géométrie arbitraire, y compris la géométrie procédurale
 - Augmente l'utilité pratique de l'algorithme
 - n'entraîne pas d'artefacts

Rappels sur le Kdtree

- Un arbre de partitionnement d'espace récursif.
 - Partition selon l'axe X et Y alternativement.
 - Chaque nœud interne stocke le nœud de division le long de x (ou y).
- Utilisé pour les localisations de points. k-nombre des attributs pour effectuer la recherche
- Interprétation géométrique - pour effectuer la recherche dans l'espace 2D - Arbre 2-d
- Rechercher les composants (x, y) interchangeable!

Exemples de Kdtree



Utilisation d'un Kd-tree comme une carte de photons

- Un arbre Kd est un arbre binaire qui divise l'espace k -dimensionnel selon une dimension à la fois.
- Chaque nouveau photon inséré crée un plan de séparation dans l'arbre kd (jusqu'à un niveau maximal)
- La dimension de la taille maximale de la cellule est choisie comme un plan de séparation
- La carte des photons est finalement équilibrée pour minimiser le temps de recherche

Structure du Kdtree

- Comment **balancer l'arbre** pour maintenir un bas temps de recherche?
- Comment efficacement **localiser les photons** les plus près d'un point 3D?
- Le balancement d'un kd-tree est similaire à celui d'un arbre binaire. L'opération s'effectue dans un temps $O(n \log n)$.
- À chaque niveau, il y a deux tâches à réaliser :
 - Choisir l'**axe de coupure**.
 - Choisir le **nœud de l'arbre** selon l'axe de coupure.

Balancement du Kdtree

- Le **choix de l'axe de coupure** se fait en trouvant selon quelle dimension la distance maximale entre deux points est la plus grande.
- Pour y arriver, on trouve la boîte englobante pour l'ensemble des positions de photons et on conserve la dimension selon laquelle la boîte est la plus large.
- Le **choix du nœud de l'arbre** se fait en trouvant la médiane des points (selon l'axe de coupure).
- Le choix du point médian assure que les sous-arbres de gauche et de droite auront le même nombre d'éléments.
- On construit par la suite les sous-arbres avec les deux sous-ensembles de photons créés par la coupure avec le nœud.

Balancement du Kdtree

BALANCE(L)

Entrées : Liste de photons L

Trouver la boîte englobante des positions de photons;

Choisir la dimension d selon laquelle la largeur de la boîte est maximale;

Trouver la médiane m des points selon la dimension d;

L1 tous les points inférieurs à la médiane;

L2 tous les points supérieurs à la médiane;

Noeud m;

Noeud:gauche BALANCE(L1);

Noeud:droite BALANCE(L2);

retourner Noeud;

Recherche des plus proches voisins

- L'**algorithme de recherche** commence à la racine de l'arbre et ajoute les photons à une liste s'ils sont à l'intérieur d'une certaine distance d .
- Pour les n voisins les plus proches, **la liste est triée** de manière à ce que le photon le plus éloigné puisse être éliminé si un meilleur choix est trouvé.
- Pour stocker la liste, on utilise une **file de priorité** (priority queue).
- Cette structure permet trois opérations :
 - Insérer un élément ;
 - Supprimer l'élément ayant la plus grande clé ;
 - Tester si la file est vide.
- Si la liste est pleine, on peut mesurer la distance entre le point d'intérêt $(x; y; z)$ et la racine de la liste (le plus proche voisin le plus loin) pour **ajuster la distance maximale** d .

Rendu de l'image

- Comment combiner l'utilisation d'une carte de photons au lancer de rayon pour obtenir un rendu de qualité supérieure.
- Séparer l'équation du rendu en plusieurs composantes qui peuvent être évaluées indépendamment.

- Pour un point x sur une surface, on évalue la luminance sortante comme étant **la somme de la luminance émise et de la luminance réfléchie** :

$$L_o(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o)$$

- Où

$$L_r(x, \omega_o) = \int_{\Omega} f_r(x, \omega_o, \omega_i) L_i(x, \omega_i) \cos \theta_i d\omega_i$$

Equation du rendu

La BRDF peut être perçue comme **la somme de deux composantes**, une diffuse ($f_{r;D}$) et une spéculaire ($f_{r;S}$) :

$$f_r(X, \omega_o, \omega_i) = f_{r,S}(X, \omega_o, \omega_i) + f_{r,D}(X, \omega_o, \omega_i)$$

Le terme $L_i(x; \omega_i)$ représentant la **luminance entrante** peut être séparé en trois composantes :

- $L_{i;l}$, qui est l'**illumination directe** provenant des sources de lumière ;
- $L_{i;c}$, qui représente les **caustiques**, c'est-à-dire de l'illumination indirecte reçue via réflexion spéculaire ou réfraction.
- $L_{i;d}$, qui représente l'**illumination indirecte** reçue par réflexion diffuse

$$L_i(X, \omega_i) = L_{i,l}(X, \omega_i) + L_{i,c}(X, \omega_i) + L_{i,d}(X, \omega_i)$$

Equation du rendu

$$\begin{aligned} L_r(x, \omega_0) &= \int_{\Omega} f_r(x, \omega_0, \omega_i) L_i(x, \omega_i) \cos \theta_i d\omega_i \\ &= \int_{\Omega} f_r(x, \omega_0, \omega_i) L_{i,l}(x, \omega_i) \cos \theta_i d\omega_i + \\ &\quad \int_{\Omega} f_{r,s}(x, \omega_0, \omega_i) (L_{i,c}(x, \omega_i) + L_{i,d}(x, \omega_i)) \cos \theta_i d\omega_i + \\ &\quad \int_{\Omega} f_{r,D}(x, \omega_0, \omega_i) L_{i,c}(x, \omega_i) \cos \theta_i d\omega_i + \\ &\quad \int_{\Omega} f_{r,D}(x, \omega_0, \omega_i) L_{i,d}(x, \omega_i) \cos \theta_i d\omega_i \end{aligned}$$