M2-Images

Transport et Monte Carlo

J.C. Iehl

October 20, 2020

résumé des épisodes précédents...

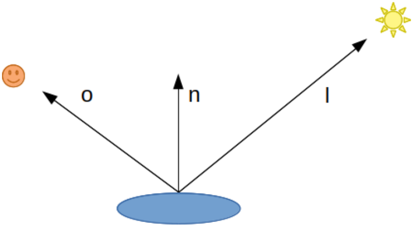
- formulation du transport de la lumière,
- sur une direction,
- intégration nécessaire sur un ensemble de directions,
- estimateur de Monte Carlo,
- variable aléatoire et densité de probabilité...

et pendant ce temps...

formulation du transport :

- lumière émise par (un point sur) une source,
- réfléchie par un point sur une surface,
- mesurée par un point du plan image / un pixel.
- ou de manière équivalente :
  un point p sur une surface est éclairé par une source dans la direction $\vec{l}$ et observé par la camera dans la direction $\vec{o}$.

1 point / 1 direction



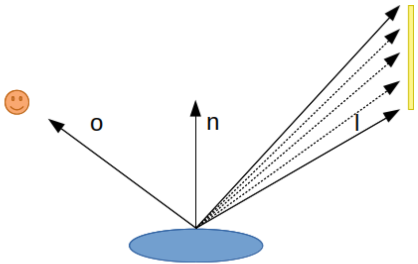
$$L_r(p, \vec{o}) = L_i(p, \vec{l}) f_r(\vec{l}, p, \vec{o}) cos\theta$$

éclairage par une source

une source de lumière :

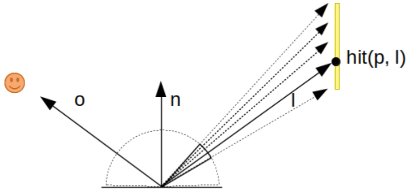
- est composée de plusieurs points,
- visibles par un ensemble de directions depuis le point p,
- il faut sommer la lumière *transportée* par chaque direction pour obtenir le résultat complet, cf intégration.

ensemble de points / ensemble de directions



$$L_r(p, \vec{o}) = \int L_i(p, \vec{l}) f_r(\vec{l}, p, \vec{o}) cos\theta dl$$

integration : ensemble de directions

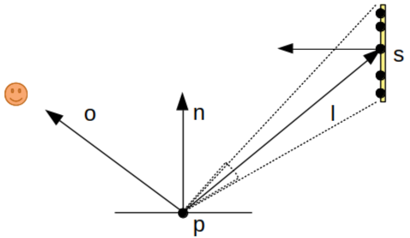


$$L_r(p, \vec{o}) = \int_{\Omega} L_i(p, \vec{l}) f_r(\vec{l}, p, \vec{o}) cos\theta dl$$

$$L_r(p, \vec{o}) = \int_{\Omega_{source}} L_e(hit(p, \vec{l}), -\vec{l}) f_r(\vec{l}, p, \vec{o}) cos\theta dl$$

integration : ensemble de points

mais on peut viser la source de lumière :



$$L_r(p, \vec{o}) \quad = \quad \int_{source} L_e(s, \vec{sp}) f_r(\vec{ps}, p, \vec{o}) \frac{V(p,s) cos\theta_s}{d(p,s)^2} cos\theta ds$$

intégration numérique

rappel : estimateur Monte Carlo

$$I = \int f(x)dx \equiv \int \frac{f(x)}{p(x)} p(x)dx = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

- x variable aléatoire de densité $p(x)$:
- espérance de x : $\int xp(x)dx = \frac{1}{N} \sum_{i=1}^{N} x_i$
- espérance de $f(x)$: $\int f(x)p(x)dx = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$
- *rappel :* une variable aléatoire est toujours associée à une fonction de densité p

et pendant ce temps...

exemple d'integration :

- $I = \int_\Omega \cos\theta d\omega$
- $I = \frac{1}{N}\sum_i \cos\theta_i \frac{1}{pdf(\omega_i)}$
- ω est la variable aléatoire
- on connait au moins 2 manières de générer des directions ω sur l'hemisphere,
- cf GI compendium, eq 34 et 35

exemple

```
#include "vec.h"
// genere une direction sur l'hemisphere,
// cf GI compendium, eq 34
Vector sample34( const float u1, const float u2 ) {
    // coordonnees theta, phi
    float cos_theta= u1;
    float phi= float(2 * M_PI) * u2;

    // passage vers x, y, z
    float sin_theta= std::sqrt(1 - cos_theta*cos_theta);
    return Vector( std::cos(phi) * sin_theta,
            std::sin(phi) * sin_theta,
            cos_theta );
}

// evalue la densite de proba, la pdf de la direction
float pdf34( const Vector& w ) {
    if(w.z < 0)  return 0;
    return 1 / float(2 * M_PI);
}
// remarque : on pourrait aussi ecrire une fonction qui renvoie une
     direction et sa pdf...
```

exemple

```
// generateur materiel
std::random_device seed;

// initialise le generateur de nombres aleatoires.
// (random genere des entiers non signes 32 bits)
std::default_random_engine random(seed());

// normalise les nombres aleatoires entre 0 et 1
std::uniform_real_distribution<float> u01(0, 1);
```

exemple

```
int n= 1024;
float I1= 0;
for(int i= 0; i < n; i++) {
    // genere 2 nombres aleatoires entre 0 et 1
    float u1= u01(random);
    float u2= u01(random);

    // genere une direction aleatoire, utilise 2 nombres aleatoires
    Vector w= sample34(u1, u2);
    float p= pdf34(w);

    I1+= f(w) / p;
}
I1/=float(n);
printf("N=␣%d␣I1=␣%f\n", n, I1);
```

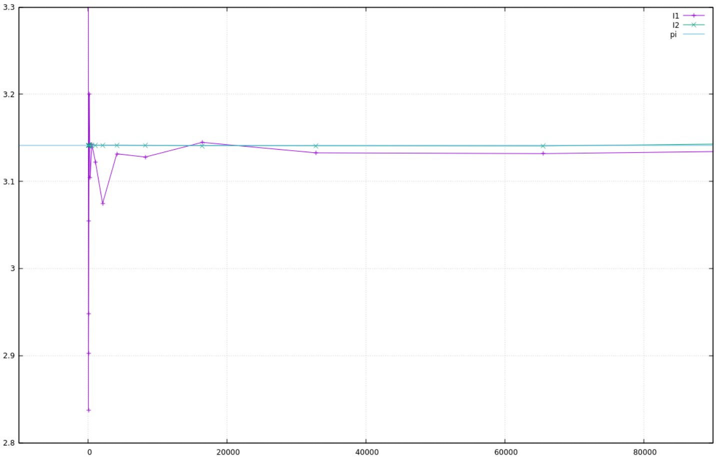
exemple : et avec l'eq 35

```
#include "vec.h"
// genere une direction sur l'hemisphere,
// cf GI compendium, eq 35
Vector sample35( const float u1, const float u2 ) {
    // coordonnees theta, phi
    float cos_theta= std::sqrt(u1);
    float phi= float(2 * M_PI) * u2;

    // passage vers x, y, z
    float sin_theta= std::sqrt(1 - cos_theta*cos_theta);
    return Vector( std::cos(phi) * sin_theta,
        std::sin(phi) * sin_theta,
        cos_theta );
}

// evalue la densite de proba, la pdf de la direction
float pdf35( const Vector& w ) {
    if(w.z < 0)  return 0;
    return w.z / float(M_PI);
}
// remarque : on pourrait aussi ecrire une fonction qui renvoie une
     direction et sa pdf...
```

exemple



et alors ?

on dirait qu'une solution est plus précise que l'autre ??

pourquoi ?

- eq 35 : la pdf est proportionnelle à la fonction intégrée...
- 1 seul échantillon est nécessaire pour obtenir le bon résultat...
- mais on n'est pas toujours dans ce cas...

et il faut *beaucoup* d'échantillons pour obtenir une valeur très précise... ($N < 100000$) sur le graphe.

éclairage direct : 1 source

écrire les estimateurs Monte Carlo :

- integration : ensemble de directions, variable $\vec{l}$, densité $pdf(\vec{l})$

$$L_r(p, \vec{o}) = \frac{1}{N} \sum_{i=1}^{N} L_e(hit(p, \vec{l_i}), -\vec{l_i}) f_r(\vec{l_i}, p, \vec{o}) cos\theta_i \frac{1}{pdf(\vec{l_i})}$$

- integration : ensemble de points, variable s, densité $pdf(s)$

$$L_r(p, \vec{o}) = \frac{1}{N} \sum_{i=1}^{N} L_e(s_i, \vec{s_i p}) f_r(\vec{p_i s}, p, \vec{o}) \frac{V(p, s_i) cos\theta_{si}}{d(p, s_i)^2} cos\theta_i \frac{1}{pdf(s_i)}$$

et alors ?

prochaine étape :

- générer des points s ou des directions $\vec{l}$...
- cf recueil de formules : GI Compendium
- ou PBRT pour plus de détails et d'exemples.

et alors ?

l'algo est toujours le même :

- générer un échantillon x
  et évaluer sa densité de proba : $pdf(x)$,
- évaluer la fonction à intégrer : $f(x)$,
- ajouter au total : $I = I + f(x)/pdf(x)$,
- recommencer N fois,
- $I = I/float(N)$,
- fini !!

éclairage direct : exemple

on veut calculer : ensemble de directions, variable $\vec{l}$, densité $pdf(\vec{l})$

$$L_r(p, \vec{o}) = \int_\Omega L_e(hit(p, \vec{l}), -\vec{l}) f_r(\vec{l}, p, \vec{o}) cos\theta dl$$

avec $L_e(x, \vec{v}) = 0$ si le point x n'est pas sur une source de lumière.

éclairage direct : exemple

on écrit l'estimateur Monte Carlo :

$$L_r(p, \vec{o}) = \frac{1}{N} \sum_{i=1}^{N} L_e(hit(p, \vec{l_i}), -\vec{l_i}) f_r(\vec{l_i}, p, \vec{o}) cos\theta_i \frac{1}{pdf(\vec{l_i})}$$

éclairage direct : exemple

algo :

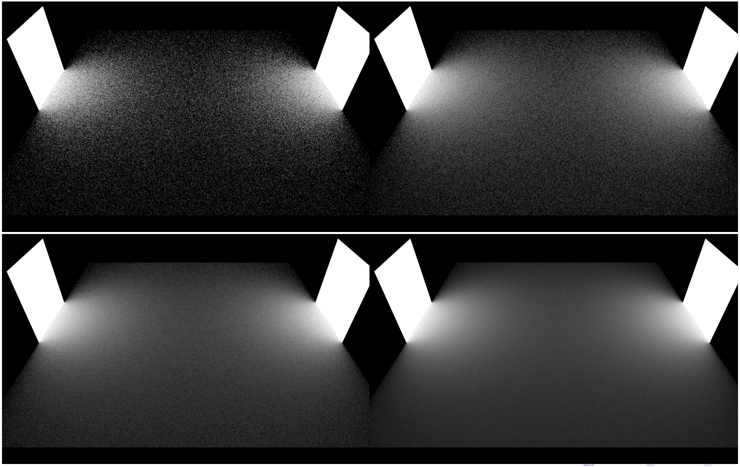
- générer une direction $\vec{l_i}$
  et évaluer $pdf(\vec{l_i})$, cf GI Compendium eq 34, ou eq 35
- évaluer $hit(p, \vec{l_i})$: construire un rayon $(p, \vec{l_i})$
- si intersection, on connait $hit(p, \vec{l_i})$ et sa matière,
- $L_e(hit(p, \vec{l_i}), -\vec{l_i}) = \texttt{matiere.emission}$,
- on connait la valeur de tous les termes,
- $I = I + f(\vec{l_i})/pdf(\vec{l_i})$
- recommencer N fois,
- renvoyer le résultat I/N.

éclairage direct : exemple

et voila !

- dernier détail : les directions générées sont dans un repère local,
- passer dans le repère de la scene...
- cf utilitaire `World` dans le tp.

éclairage direct : résultats



éclairage ambiant

c'est la même chose...

- écrire l'estimateur à partir de la formulation,
- identifier la variable aléatoire et les densités utilisables,
- estimer la moyenne de l'estimateur pour N valeurs...

il y a au moins 2 densités utilisables : (eq 34) $\frac{1}{2\pi}$ et (eq 35) $\frac{\cos\theta}{\pi}$.
quelle différence ? faites le test.

vous pouvez aussi comparer avec la spirale de Fibonacci perturbée...

éclairage direct : plusieurs sources

surprise !!

- c'est encore la même chose !!
- mais il y a une nouveauté :
- la variable aléatoire est composée de 2 parties indépendantes :
- choisir une source, S, densité $p(S)$
- choisir un point s, densité $p(s)$ sur la source sélectionnée S, densite $p(S)$

les surprises les plus courtes sont les meilleures ?
(ou pas...)

- par exemple :
- choix uniforme d'une source S parmi 4 sources : $p(S) = 1/4$,
- choix d'un point s sur la source : $p(s) = 1/aire(S)$
- quelle est la densité de la variable complete : choisir la source puis choisir le point ?
- les 2 sont indépendantes : $p(S)p(s) = \frac{1}{4}\frac{1}{aire(S)}$

peut on choisir la source d'une autre manière ?

indication : regardez le chapitre 13.3.1 de PBRT.
peut-on utiliser la méthode d'inversion pour sélectionner les sources ?