

SYNTHÈSE D'IMAGES

TP3 : Lancé de rayons avec POV-Ray

réalisé par LE Viet Man - promotion 15

FONCTIONNEMENT DU PROGRAMME

Usage du programme

Ce programme a pour but de construire une image photoréaliste avec POV-Ray. Pour lancer le programme, on peut utiliser la commande suivante :

```
povray res[type] unevue
```

où

- res : indique le fichier de configuration res.ini dans lequel j'ai défini cinq résolutions que l'utilisateur peut choisir pour construire l'image.
- type : y compris Low (80x60), Med (320x200), High (640x480), Really High (800x600) et Normal (120x100). Avec la résolution Normal, on peut ignorer la partie [type], c'est-à-dire on peut taper seulement la partie res.
- unevue : indique le fichier de configuration unevue.ini de ce programme.

L'image de résultat est sous type de PNG, est nommée unevue, même nom avec le fichier pov et ini.

Par ailleurs, en changeant la valeur des variables dans la partie Contrôle du code, l'utilisateur peut obtenir de différentes images. Le programme soutient des options suivantes :

```
#declare Camera = 3; // 0,1,2,3,4,5,6,7,8
```

```
#declare Sun = on; // on, off
```

```
#declare LightLamp = on; // on, off
```

```
#declare LightCandle = on; // on, off
```

```
#declare Radiosity = off; // on, off
```

```
#declare Photons = off; // on, off
```

```
#declare Atmospheric = on; // on, off
```

```
#declare show_GlassWindow = true; // true, false
```

- Camera : le programme soutient neuf positions différentes de camera. 0 - regarde vers le coin de salle ayant la fenêtre. 1 - regarde l'image de moi et ma femme. 2 - regarde la fleur et le vase de vin. 3 - regarde de façons flou la fleur et le vase de vin. 4 - regarde la bougie. 5 - regarde vers le coin de salle ayant le bahut et la table basse. 6 - regarde l'échiquier. 7 - regarde du centre de la salle vers le premier coin. 8 - regarde du centre de la salle vers le deuxième coin.

- Sun : pour l'objectif de changer entre le matin et la soirée
- LightLamp : on - allumer la lampe, off - éteindre la lampe
- LightCandle : on - allumer la bougie, off - éteindre la bougie
- Radiosity : on - utiliser la méthode de la radiostie, off - à l'inverse
- Photons : on - utiliser la méthode Photon, off - à l'inverse
- Atmospheric : on - créer l'effet atmosphérique, off - à l'inverse
- show_GlassWindow : true - la fenêtre avec la vitre, false - pas de vitre

Quelques exemples

- Pour construire l'image de résolution de 320x200, on utilise la commande suivante :

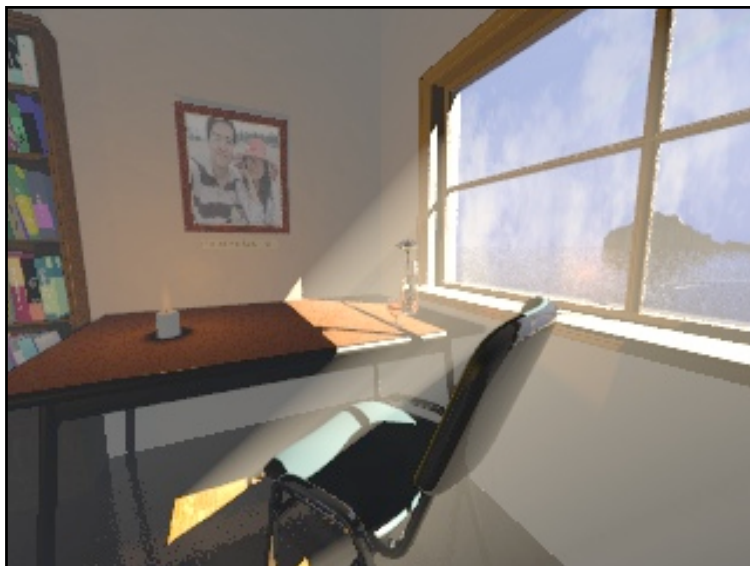
`povray res[Med] unevue`

- Pour construire l'image de résolution de 120x100, on utilise la commande suivante :

`povray res unevue`

MODELISATION DE LA SCENE

J'ai construit une salle dans laquelle il y a deux coin important. Au premier coin, on peut voir une table, une chaise, une fleur, une bougie, une image, une bibliothèque, une fenêtre où on peut voir la mer avec une île et un arc-en-ciel. Au deuxième coin, on peut voir une table basse avec une boule en verre et un échiquier, une image de paysage, un bahut avec trois bouteilles de vin et une lampe. Deux images suivantes démontrent deux coins de la salle :



Le premier coin de la salle



Le deuxième coin de la salle

Extérieur

A l'extérieur de la salle, j'ai construit des objets suivants :

- *Le ciel* : j'ai utilisé l'objet `sky_sphere` avec deux types de pigment. Le premier pigment utilise la forme de vague `poly_wave` avec un `color_map`. Le deuxième pigment utilise la

texture de type bozo et turbulence = 0.65 pour construire le nuage. Ces nuages sont peu réaliste, mais il rend le temps de calcul rapide. J'ai testé la technique de génération des nuages (<http://www.oyonale.com/modeles.php?lang=en&page=36>), mais le temps de calcul est trop grand.

- *L'arc-en-ciel* : j'ai utilisé l'objet rainbow avec angle = 35 et arc_angle = 100.
- *La mer* : j'ai utilisé la mer d'exemple Balcony de POV-Ray où on utilise la fonction f_ridged_mf pour créer le moutonnement dans la partie normal. C'est une technique simple et efficace pour mon scène. Dans la partie finish, on configure les valeurs reflection pour permettre l'effet de réflexion. Dans la partie interior, on utilise la valeur ior = 1.31 (c'est la valeur de l'eau) et la Light Fading. On utilise encore un objet plane pour construire le plancher de la mer avec les valeurs ambient = 0 et diffuse = 0.
- *L'île* : j'ai utilisé un objet height_field pour construire une île avec une color_map de deux couleurs blanc et noir pour créer des buttes de terre blanches et noires. En utilisant le modificateur de pattern warp, on peut construire la terre mouvementée. J'ai également utilisé deux textures pour l'herbe et la pierre.
- *Le brouillage* : J'ai ajouté l'effet de brouillage dans la scène pour rendre l'horizon plus réaliste. Ce brouillage est de type « brouillage sur le sol » avec fog_type = 2, distance = 30 pour créer le ciel vapoureux.

Salle

Pour construire la salle, j'ai écrit un macro où j'ai utilisé un opérateur CSG union pour intégrer les murs, le plafond, le plancher. Pour créer le trou pour la porte et la fenêtre, j'ai utilisé un autre opérateur CSG difference (doublé dans l'opérateur union). Les murs et le plafond sont configurés comme granite dans la partie normal. Le plancher possède les textures en créant un image_map de l'image parquet.jpg. La surface du plancher est aussi créée par cette image avec le paramètre bump_map. J'ai aussi configuré ce plancher avec le paramètre reflection {0, 0.4} dans la partie finish pour permettre l'effet de réflexion dans ce plancher.

J'ai utilisé le code source de la fenêtre dans l'image Cacophony de Phil Brewer (<http://www.irtc.org/stills/2006-04-30.html>) pour construire une fenêtre. Cette fenêtre possède un cadre construit par un objet prism de type cubic_spline et un verre construit par un objet box avec la paramètre hollow pour créer la transparence. Au début, j'ai utilisé la paramètre ior 1.5 pour la transparence, mais ensuite j'ai utilisé le media de type Scattering. Cette configuration est aussi pour créer l'effet de l'atmosphère. J'ai configuré les valeurs ambient = 0.1, diffuse = 0.1 pour créer la transparence dépolie et sans doute reflection = 0.1 pour l'effet de réflexion.

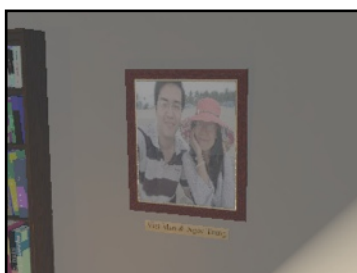
Intérieur - premier coin

Au premier coin, il y a sept objets suivants :

- *Un siège* : j'ai utilisé le code source du siège dans l'image Conference chaire (<http://www.oyonale.com/modeles.php?lang=en&page=46>). Ce siège est construit par des objets de

type mesh2. Le cadre du siège est en plastique avec les valeurs de reflection = {0, 0.4} pour avoir la capacité de réflexion.

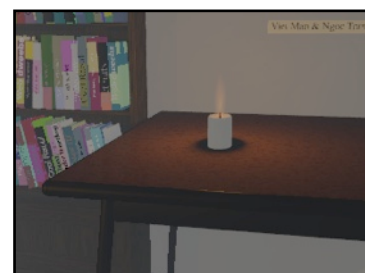
- *Une table* : quatre pieds de la table est construit par l'opérateur CSG union des objets de type cone. Le châssis dessous la surface de la table est aussi construit par l'opérateur CSG union des objets de types sphere et cylinder. Ces pieds et le châssis est vernissés avec le paramètre specular = 1 dans la partie finish. Le surface de la table est construit par un objet de type box avec le matériau qui est mélangé entre le marbre et le bois. Ce surface est peu reflex avec la paramètre reflection 0.01.
- *Une bibliothèque* : j'ai utilisé la méthode de génération une bibliothèque dans <http://www.oyonale.com/modeles.php?lang=en&page=19>. On fournit trois macros pour créer automatiquement une bibliothèque dont la position et l'inclinaison des livres sont calculées par hasard, le titre et le font des livres sont pris aléatoirement en chargeant les données dans deux fichiers title.txt et font.txt. La texture de bibliothèque est wood et son pattern est modifié par turbulence 0.2.
- *Une image* : j'ai construit une image qui est ma photo, suspendue sur le mur. J'ai utilisé le macro Picture de Realistic Painting de Gail Shaw (<http://www.irtc.org/stills/2002-02-28.html>) pour construire cette image. Le cadre de l'image est construit par deux opérateurs CSG doublés difference, union et des objets de type box, cylinder. J'ai utilisé la texture du bois (T_Wood6) pour le cadre de la photo. L'image est construit par un box avec la texture de type image_map en chargeant ma photo. J'ai configuré les paramètres specular=1 et reflection=0.25 pour créer le sentiment qu'il y a un miroir en face de l'image. Mais cette méthode n'est pas bien parce que elle rend l'image flou. Il faut utiliser un autre box en face le box de l'image avec la texture de verre. J'ai aussi écrit mon nom sur une carte jaune collée sous l'image.
- *Un vase de fleur* : j'ai utilisé le code source de l'image Flower de Bernard Hatt (<http://www.irtc.org/stills/2005-08-31.html>). Tout d'abord, le vase est construit par trois opérateurs difference, merge et union avec trois types de objets cylinder, torus et sphere. Ce vase est transparent avec le texture F_Glass7 et ior 1.525 dans la partie interior. Il est aussi un source d'émission de lumière, donc j'ai configuré trois paramètres fade_distance (10), fade_power (2) et fade_colour. L'eau dans le vase est construit par trois opérateurs union, difference et intersection et trois types d'objets torus, sphere et cylinder. Le code crée aussi des bulles de l'air à l'intérieur de l'eau. J'ai configuré l'effet de réflexion de l'eau



Ma photo



Vase de fleur et un verre de vin



La bougie

avec la paramètre reflection {0.01, 1 fresnel on}. L'eau ont aussi la capacité de réfraction parce que j'ai utilisé face_distance (10), face_power (2) et fade_colour.

- *Un verre de vin* : j'ai utilisé le macro MakeGlass de <http://www.oyonale.com/modeles.php?lang=en&page=13> pour construire un verre de vin. Ce verre est construit par une combinaison entre l'opérateur union et deux lathes de type cubic_spline. Le verre est transparent (ior 1.5) et possède seulement la capacité de réflexion (reflection {0.01, 0.3}). En semblablement, le vin possède la texture C_Liquid avec la capacité de réflexion (reflection {0.01, 0.3 fresnel on}) et la capacité de réfraction (fade_distance 2, fade_power 2).
- *Une bougie* : j'ai utilisé le macro Candle_2 (http://www.f-lohmueller.de/pov_tut/objects/household/Candle_2.inc) pour construire cette bougie. En forme, on utilise deux opérateurs union, difference et un deux objets sphere et Round_Cylinder pour le tronc de la bougie, un objet cylinder pour la cire et un opérateur union et deux objets cylinder pour la mèche.

Intérieur - deuxième coin

Au deuxième coin, il y a aussi sept objets suivants :

- *Un tapis* : j'ai utilisé un objet de type box pour construire ce tapis. Il a la texture de type image_map en chargeant l'image carpet.png.
- *Une table basse* : j'ai utilisé le code source de <http://www.oyonale.com/modeles.php?lang=en&page=31> pour la construire. Quatre pieds sont construits par l'objet lathe de type cubic_spline et le surface de la table est un objet superellipsoid. Cette table utilise la texture granite.
- *Un bahut* : j'ai utilisé le code source de Ghost Light de Bill Pragnell (<http://www.irtc.org/stills/2006-10-31.html>) pour construire ce bahut et trois bouteilles de vin. Ce bahut est créé par des objets Round_Box et possède la texture granite avec un peu de réflexion (reflection 0.01), à l'exception des boutons de porte qui ont la valeur de réflexion de 1 (reflection 1). Trois bouteilles de vin est créées par des objets lathe de type bezier_spline. Les bouteilles sont configurées la transparence et la réflexion avec les paramètres reflection {0.2 exponent 0.8 } et ior 1.5. Le vin est simple avec la transparence, la valeur ior est 1.33.
- *Une image* : j'ai utilisé même objets de l'image au premier coin, mais la texture du box de contenue charge l'autre image.
- *Un échiquier* : j'ai utilisé l'exemple GlassChess de POV-Ray sans la modification dans la forme et la texture.
- *Une boule en verre* : elle est simplement construit par un opérateur CSG union de deux objets Round_Box pour un piédouche et sphere pour la boule. Le piédouche a la texture granite et un peu de réflexion (reflection {0.2 exponent 0.8}). La boule est comme un miroir qui peut réfléchir les objets autour de lui. Donc, la valeur de réflexion est 1 (reflection 1).

Lumière

Dans ce scène, j'ai utilisé trois sources de lumières : un soleil, une bougie et une lampe. Le soleil est une source Pointlight avec la couleur blanc laiteux. La bougie et la lampe sont deux sources Arealight avec la couleur orange. Pour la lampe, j'utilise un objet de type sphere pour modéliser une ampoule ronde. Pour la bougie, il est plus complexe, j'utilise d'abord un opérateur CSG merge et deux opérateurs CSG intersection pour créer la forme de la flamme, suite j'utilise la fonctionnalité Media de type Emitting pour créer un environnement additif qui peut absorber la couleur des sources de luminaire. Ces deux sources utilisent encore la technique Light Fading pour créer le réalisme dans le scène avec 0.3 pour fade_distance et 2 pour fade_power.

Atmosphère

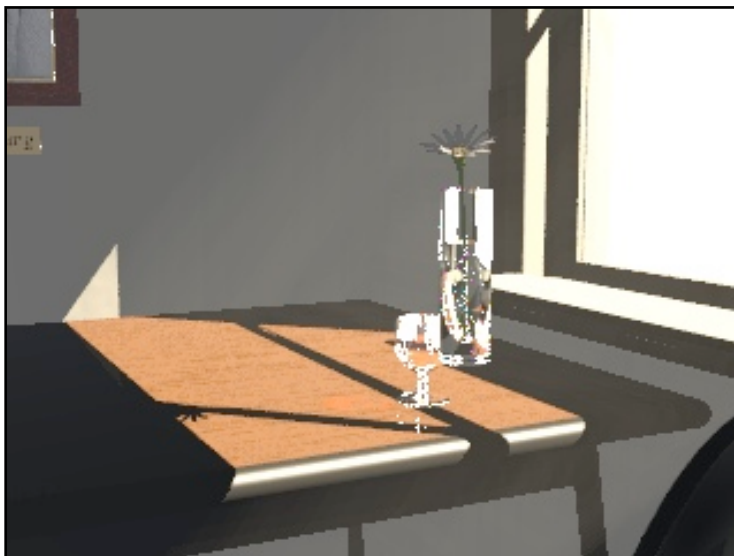
Dans ce TP, pour relever le réalisme de l'image, j'ai testé deux méthodes de Atmospheric Media. La première méthode, j'ai seulement ajouté le code suivant au scène sans l'ajout aux n'importe quels objets :

```
media { // atmospheric media sample
  intervals 10
  scattering { 1, rgb 0.03 }
  samples 1, 10
  confidence 0.9999
  variance 1/1000
  ratio 0.9
}
```

La deuxième méthode, j'ai ajouté un objet box vide (hollow) à l'intérieur de la salle qui ont l'instruction Media suivante :

```
media {
  scattering { 1, 0.17 extinction 0.01 }
  samples 100, 500
}
```

Voici les résultats de deux types :



La première méthode



La deuxième méthode

Il est facile à trouver que la deuxième méthode est plus parfaite que la première. La deuxième méthode montre des fautes sur les surfaces en verre ou transparents. J'ai utilisé la deuxième méthode pour tous les images suivantes.

LANCER DE RAYONS

Voici les résultats après avoir appliqué la lancé de rayons et l'atmosphère avec les positions différentes de caméra :



Caméra 0



Caméra 1



Caméra 2



Caméra 4

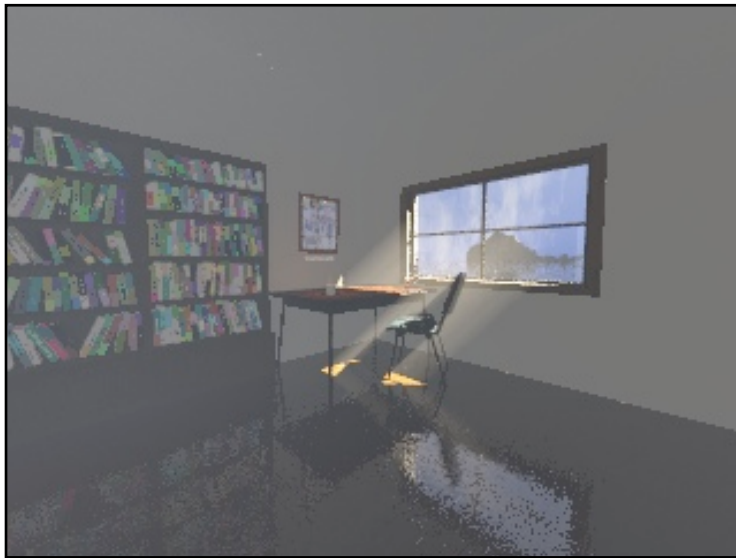
Avec le lancer de rayons, le résultat est très bon, par exemple l'ombre des objets; l'ombre des objets sur la fenêtre, sur les images et sur la boule en verre; les réflexions spéculaires sur le vase, le verre de vin; l'ombre de la bougie; les rayons de lumière du soleil; la luminescence faible de la bougie; les rayons de la lampe sur le mur, la table basse et le bahut. Pourtant, les réflexions diffuses sur les murs, la fenêtre, la table et l'effet de caustique sont moins bien. De plus, je trouve un autre problème. C'est que l'effet de polka dots. Je pense que le media atmosphérique ou la grande et petite distance entre des sources de lumière.



Caméra 5



Caméra 6



Caméra 7



Caméra 8

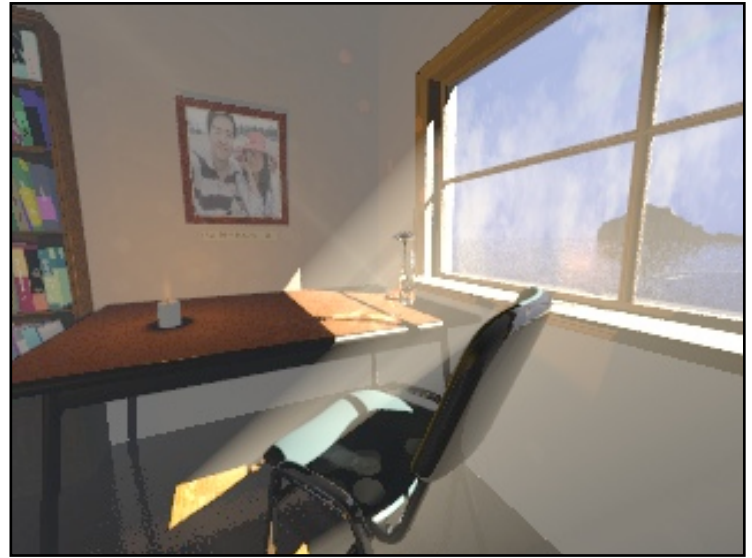
PHOTONS

Pour créer l'effet caustique au scène, j'ai utilisé la méthode Photons. Tout d'abord, j'ai configuré la méthode Photons au scène avec les instructions suivantes :

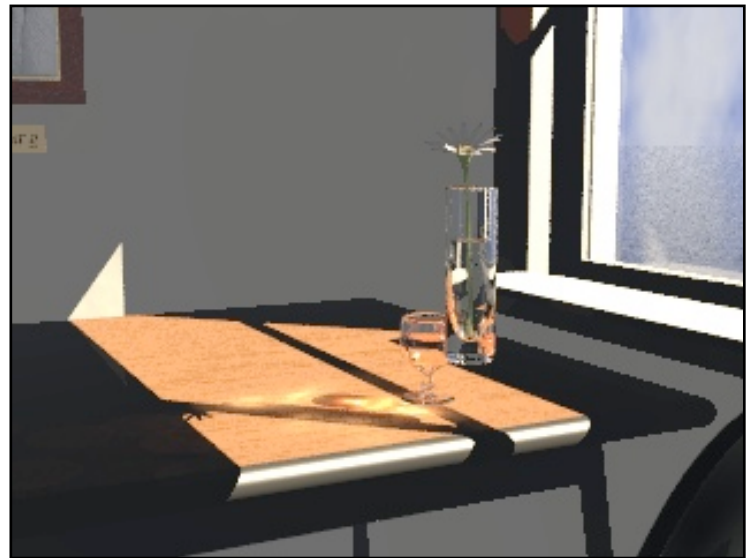
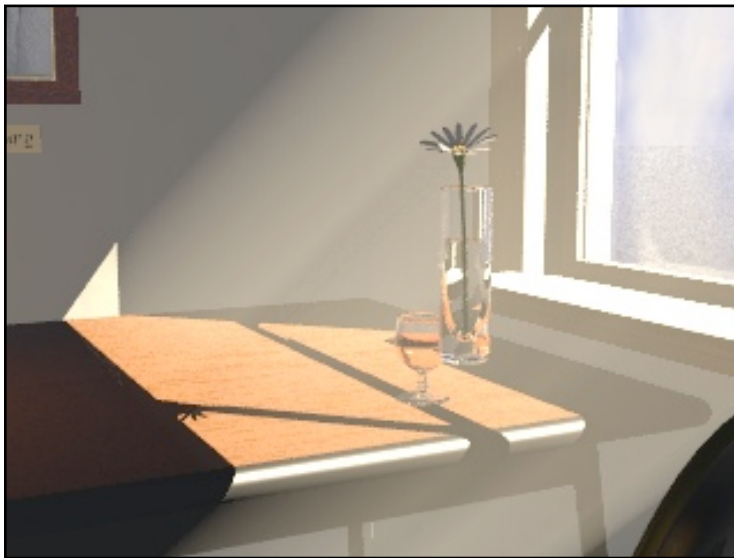
```
photons {
    spacing 0.002
    media 100
}
```

La deuxième instruction (`media 100`) pour permettre l'interaction entre les photons et les medias. Ensuite, pour le box vide à l'intérieur de la salle, j'ai ajouté l'instruction `photons {pass through}` pour que les photons peuvent traverser dans le box. Pour les sources de lumière, j'ai étendu l'effet de réflexion (`photons {reflection off}`). Enfin, dépendant le caractéristique de chaque objet, j'ai ajouté et étendu ou allumé l'effet de réflexion et de réfraction.

Voici les résultats de la méthode Photons :



Caméra 0 - l'image à droit : utiliser la méthode Photons



Caméra 2 - l'image à droit : utiliser la méthode Photons

Je trouve que le résultat de caustiques de réfraction et de réflexion est meilleur que le lancer de rayons. La fenêtre réfléchit bien les rayons lumineux et les verres focalisent bien la lumière.

RADIOSITE

La radiosité permet de rendre plus réaliste l'image. J'ai utilisé les paramètres suivants pour la radiosité pour améliorer ma scène :

```
radiosity{
    pretrace_start 0.08
    pretrace_end   0.01
    count 130
    nearest_count 5
    error_bound 0.3

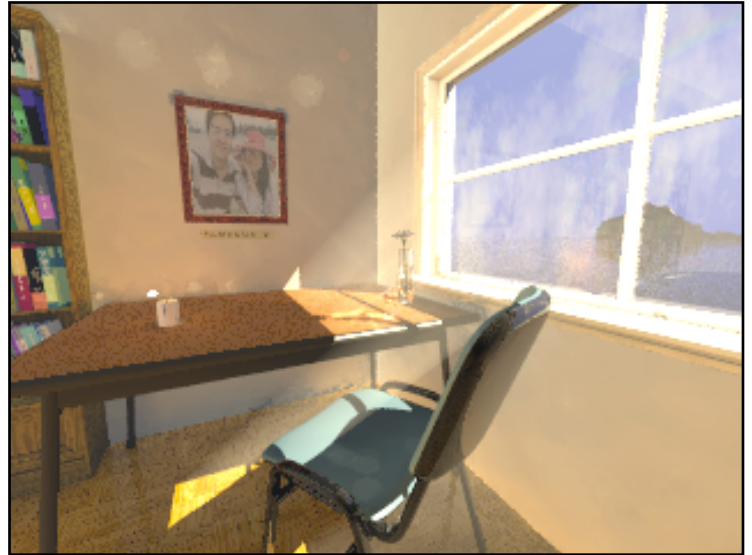
    recursion_limit 1
    low_error_factor 0.5
    gray_threshold 0.0
}
```

```

    minimum_reuse 0.015
    brightness 1.0
    adc_bailout 0.01/2
    normal on
    media on
    always_sample off
}

```

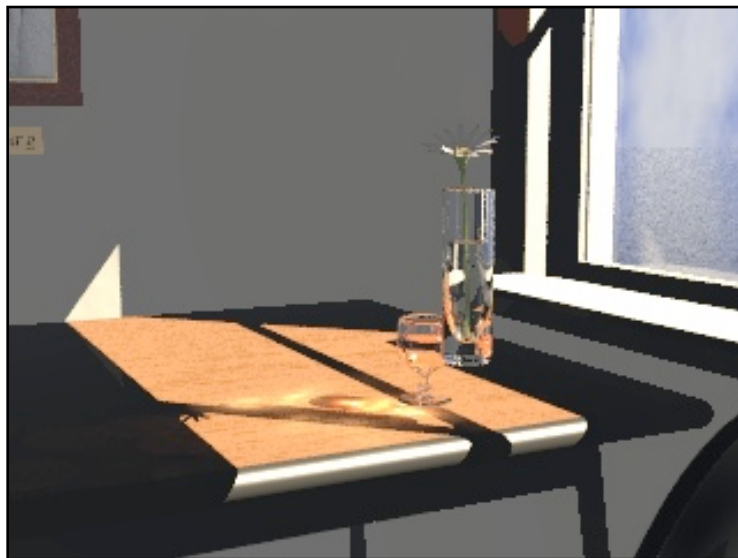
La valeur brightness est 1 pour rendre la salle éclairée. Voici les résultats de la méthode Radiosité :



Caméra 0 - l'image à gauche utilise la méthode Photons et l'image à droite utilise la méthode Radiosité



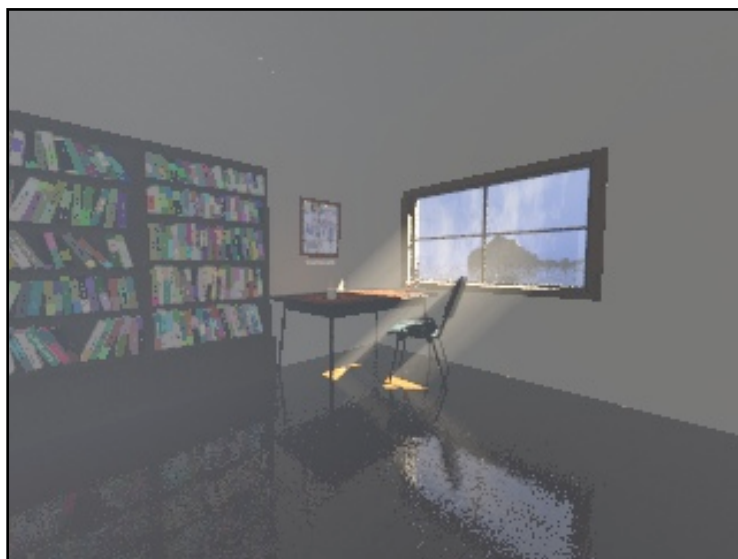
Caméra 1 - l'image à gauche utilise la méthode Lancer de rayons et l'image à droite utilise la méthode Radiosité et Photons



Caméra 2 - l'image à gauche utilise la méthode Photons de rayons et l'image à droite utilise la méthode Radiosité



Caméra 5 - l'image à gauche utilise la méthode Photons de rayons et l'image à droite utilise la méthode Radiosité



Caméra 7 - l'image à gauche utilise la méthode Lancer de rayons de rayons et l'image à droite utilise la méthode Radiosité et Photons

On peut voir qu'avec radiosit , l'image est plus claire et plus r aliste, elle est aussi plus lissage parce qu'avec seulement le lanc  de rayons, la lumi re de chaque pixel est calcul e en seulement tenant compte la lumi re   partir des rayons des sources de lumi res tandis la radiosit  tenant compte aussi les interactions des lumi res entre les objets dans la sc ne, un objet obtient aussi les lumi res   partir des autres objets (par r flexion). Pourtant, je trouve aussi l'effet de polka dots sur quelques images et les r gions noires.

Par ailleurs, quand j'utilise la combinaison avec la m thode Radiosit  et Photons, le temps de calcul de mon sc ne est trop grande sur le syst me d'exploitation Windows, mais le r sultat est meilleur et tr s lisse. Pourtant, dans le syst me d'exploitation Ubuntu, le temps de calcul est moins grande mais il y a des polka dots et des r gions noires.

CONCLUSION

La m thode Lanc  de rayons et l'outil POV-Ray est une bonne m thode pour cr er les images tr s r alistes et tr s jolies. Un point tr s important est que le choix raisonnables des valeurs pour les param tres de textures, de couleurs, de r flexion et de interioir, etc. La combinaison avec la m thode Radiosit  est une bonne solution qui rend les images les plus r alistes. La m thode Photons et l'atmosph re fournissent aussi  a. Mais le temps de calcul sera trop grand.