

Manuel d'utilisation de
KF-Ray 1.0.1, raytracer parallèle

Karin Aït-Si-Amer & Florian DANG

Mai 2009

Vous êtes en possession de KF-Ray un programme qui permet de générer des images de synthèse au rendu 3D impressionnant utilisant la méthode du lancer de rayon (raytracing). Devant la complexité d'utilisation de POV-Ray, la multitude de raytracers peu documentés, pas assez complets, obsolètes, trop personnalisés, nous avons jugé intéressant de développer un raytracer actuel, KF-Ray avec une vision altruiste.

KF-Ray se veut être simple d'utilisation en incluant une interface graphique, tout en ayant à disposition des fonctionnalités puissantes et avancées en particulier sur le parallélisme. KF-Ray a un code explicite documenté par Doxygen et se montre donc didactique pour tout curieux souhaitant programmer. Vous avez à disposition dans la sous-section 1.1.3 les features de KF-Ray.

Ce manuel d'utilisation va vous permettre de prendre en main rapidement KF-Ray et de vous faire découvrir toutes les options qu'il propose telle que la parallélisation sur plusieurs machines.

Tout d'abord, après avoir compilé KF-Ray, vous allez pour une première approche lancer le programme sans option pour générer votre première image 3D. Ensuite, vous allez apprendre à écrire des fichiers de scènes KFR pour réaliser vos propres scènes. Enfin vous pourrez générer leur rendu 3D avec le programme en console ou tout simplement avec l'interface graphique. Nous allons tout au long de ce manuel, montrer des exemples avec les deux méthodes.

Le programme étant encore supporté, vous pouvez vous reporter sur le site <http://kf-ray.googlecode.com/> pour être à jour sur l'avancement du projet. Il existe également un site français dédié au programme sur <http://kfray.free.fr>. N'hésitez pas à nous contacter si besoin.

Table des matières

1	Première approche de KF-Ray	4
1.1	Présentation succincte de KF-Ray	4
1.1.1	KF-Ray un raytracer parallèle	4
1.1.2	Features	4
1.1.3	Les fichiers sources	5
1.2	Lancer le programme pour la première fois	5
1.2.1	Système	5
1.2.2	Requis	5
1.2.3	Instructions	6
1.3	Utiliser les arguments de KF-Ray (en console “back end”)	7
1.3.1	L’option -h (help)	7
1.3.2	L’option -i (input)	7
1.3.3	L’option -o (output)	7
1.3.4	L’option -c (clean)	7
1.3.5	L’option -d : requiert ImageMagick	8
1.3.6	L’option -b (brdf)	8
1.3.7	L’option -v (view)	8
1.3.8	L’option -t (texture)	8
1.3.9	L’option -a (anti-aliasing)	9
1.3.10	L’option -l (lines) : version parallèle uniquement	9
1.4	Utiliser l’interface graphique KF-Ray GUI (“front end”)	9

2	Créer des scènes personnalisées	11
2.1	Conseils préalables	11
2.1.1	L'algorithme de ray tracing	11
2.1.2	Conseils sur les coefficients à utiliser	11
2.2	Le fichier de description KFR	11
2.2.1	Règles générales	12
2.2.2	Créer la scène	12
2.2.3	Créer une sphère	13
2.2.4	Créer un plan	13
2.2.5	Créer un matériel	13
2.2.6	Créer une source de lumière	14
3	Utilisation de KF-Ray basique et avancée	15
3.1	Exemple de rendu 3D avec KF-Ray	15
3.1.1	Fichier complet de description de scène	15
3.1.2	En ligne de commandes	16
3.1.3	Avec l'interface graphique	16
3.2	Parallélisation avec MPI	16
3.2.1	Requis	17
3.2.2	Configurer MPI	17
3.2.3	Rendre une image avec MPI	17
3.3	Options avancées	18
3.3.1	Gagner du temps lors d'une création de scène	18
3.3.2	Anti-aliasing	18
3.4	Créer des animations	18
3.4.1	Utiliser le générateur d'images avec le GUI	18
3.4.2	Pour créer une vidéo	19

Chapitre 1

Première approche de KF-Ray

Ce chapitre va vous initier à l'utilisation de KF-Ray.

1.1 Présentation succincte de KF-Ray

1.1.1 KF-Ray un raytracer parallèle

KF-Ray est un programme qui utilise la méthode du lancer de rayon (raytracing) afin de générer une image 3D très réaliste à partir d'un fichier de description de scène.

Les fichiers de descriptions sont des fichiers textes dont la syntaxe est personnalisée et propre à KF-Ray. Leur écriture est très simple et est expliquée au cours de ce manuel. Des exemples de ces fichiers sont disponibles dans les sources ou sur le site. Ces fichiers permettent de représenter la scène en trois dimensions en renseignant l'emplacement des objets (sphères, plans), des sources de lumières ainsi que leur caractéristique (couleur, intensité, texture...).

Une fois ce fichier créé, l'utilisateur peut lancer un rendu de la scène qu'il a imaginé grâce à KF-Ray. La sortie est une image au format PPM.

1.1.2 Features

Voici un rappel des possibilités qu'offre KF-Ray v1.0.x :

- Manipule les sphères et les plans
- Admet plusieurs sources de lumières ponctuelles
- Gère les réflexions, les ombrages, les réfractions (en cours transparence uniquement pour v1.0.x)
- Choix de différents modèles d'éclairage diffus, spéculaire (Lambert, Phong...)
- Gère des textures procédurales (damier, bruit de Perlin...)
- Effet de perspective, liberté de caméra
- Fonctionne avec le repère cartésien usuel (right handed coordinates)
- Fonctionne en parallèle sur plusieurs ordinateurs

- Fichiers de descriptions de scènes 3D à syntaxe simple
- Interface graphique intuitive
- Nécessite peu de bibliothèques
- Code documenté avec Doxygen, facilement suivable et reprenable
- Peut générer des images d’animation

1.1.3 Les fichiers sources

Les sources comportent deux exécutables :

- le programme principal KF-Ray “back-end” qui s’utilise en ligne de commande et peut suffire à lui seul ;
- l’interface graphique KF-Ray GUI “front-end” qui permet une utilisation plus facile et intuitive du logiciel.

Libre à vous de choisir la manière dont vous utilisez le programme. La méthode la plus “sûre” et flexible est de se passer de l’interface graphique. Cependant, si vous voulez vous familiariser facilement avec KF-Ray, si vous êtes un utilisateur peu habitué ou réticent face aux lignes de commandes, l’interface graphique se révèle être un choix judicieux. Même pour un utilisateur avancé, l’interface propose quelques options intéressantes inédites telles que la génération automatique d’une liste d’images. Finalement, il peut être avantageux d’utiliser soit l’une soit l’autre des méthodes, dépendant de vos besoins.

1.2 Lancer le programme pour la première fois

Disponible dans le fichier README.

1.2.1 Système

Vous devez disposer d’une (ou plusieurs) machines fonctionnant sur Unix/Linux à jour. Nous avons en effet à disposition un réseau d’ordinateurs fonctionnant sur Scientific Linux. Le programme a aussi été testé sur les distributions Linux Debian/Ubuntu ainsi que sur MAC OS.

La portabilité sur Windows n’est pas gérée mais doit rester possible étant donné que la majorité des bibliothèques utilisées y sont disponibles (LAM/MPI et OpenMPI font défauts à voir du côté de MP-MPICH). Il est même possible qu’une version Windows voit le jour.

1.2.2 Requis

KF-Ray a été écrit directement en C par souci de rapidité, d’optimisation, de simplicité (pas au niveau de la programmation mais au niveau des outils nécessaires à son fonctionnement). En effet, KF-Ray fait appel à peu de bibliothèques et n’utilise pas de bibliothèque graphique par exemple.

Pour la version de base, vous devez disposer seulement des outils de compilation élémentaires : un compilateur GCC (GNU Compiler Collection) à jour et les utilitaires qui vont avec (Make).

Pour utiliser l'interface graphique, il vous faudra ajouter :

- GTK+ 2.x

Pour le parser, l'interprétation du fichier de description de scène (optionnel) :

- Lexx et Yacc (Flex et Bison ont été utilisé)

Optionnellement, vous pouvez rajouter :

- ImageMagick qui permet d'utiliser l'option -d display affichant l'image à la fin du rendu.

Pour pouvoir gérer les calculs en parallèle (optionnel), il vous faudra des bibliothèques MPI (Message Passing Interface). Voici celles qui ont été utilisées et testées :

- OpenMPI

- LAM/MPI

1.2.3 Instructions

Télécharger la dernière version à jour (et ne regardez plus ce manuel s'il y en a un nouveau) disponible sur : <http://kfray.free.fr/download/>

Ouvrez une console, placez-vous dans le répertoire où se trouve l'archive `kfray-1.0.1.tar.gz` et tapez `tar -xvzf kfray-1.0.1.tar.gz` pour l'extraire.

Une fois extraite, placez vous dans le répertoire du programme en saisissant `cd kfray-1.0.1/`.

Tapez `sh autogen.sh` pour générer les fichiers de configurations et Makefiles.

Ensuite tapez `make` pour compiler les fichiers sources.

Tapez `./kfray -d` et observez le résultat !

Vous avez obtenu votre premier rendu d'image 3D avec KF-Ray en toute simplicité !

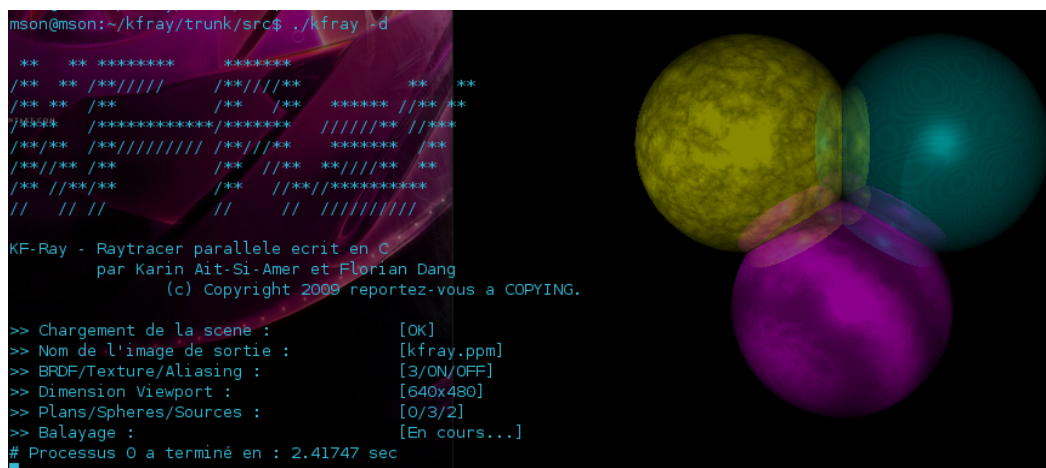


FIG. 1.1 – KF-Ray en lignes de commande (back-end)

Note : si vous n'avez pas installé ImageMagick, vous pourrez toujours voir l'image générée dans `scenes/scene1.ppm` avec un autre logiciel comme GIMP par exemple.

Regardez dans le répertoire `scenes/`, vous trouverez des fichiers au format KFR. Chacun de ces derniers décrit une scène 3D qui peut être interprété par KF-Ray. Tapez `./kfray -i scene4.kfr -d` par exemple pour avoir un rendu de la scène 4 !

1.3 Utiliser les arguments de KF-Ray (en console “back end”)

Comme tout programme Unix qui se respecte, KF-Ray gère différentes options et arguments pour son fonctionnement.

Ci-dessous un aperçu de ces options. Certaines options ne sont pas forcément nécessaires lors des premières exécutions. Vous pourrez vous y référer quand bon vous semblera.

1.3.1 L’option -h (help)

Vous avez un récapitulatif rapide des options utilisables par KF-Ray en rentrant simplement `./kfray -h` dans la console.

Cette option ne prend pas d’argument.

1.3.2 L’option -i (input)

Permet à l’utilisateur de choisir un fichier de description de scène. Le programme, si l’on ne donne pas le chemin complet du fichier, regarde dans le dossier `scenes/` si le fichier existe et procède à son rendu.

Si l’option n’est pas activée, une scène par défaut inclu dans le code source est rendu (équivalent à `scene1.kfr`).

Exemples d’utilisation : `./kfray -i scene1` ; `./kfray -i scene2.kfr`

1.3.3 L’option -o (output)

Permet à l’utilisateur de donner un nom à l’image au format PPM rendue par le programme et si l’on ne donne pas de chemin complet, l’enregistre dans le dossier `scenes/`.

Par défaut, l’image a pour nom le nom du fichier de description si l’option n’est pas activé.

Exemples d’utilisation : `./kfray -i scene2 -o img2` ; `./kfray -o image1.ppm`

1.3.4 L’option -c (clean)

Efface les images de format PPM dans le dossier `scenes/` (équivalent à la commande `rm scenes/*.ppm`).

Cette option ne prend pas d'argument.

Utilisation : `./kfray -c`

1.3.5 L'option -d : requiert ImageMagick

Affiche à la fin du rendu l'image généré par KF-Ray grâce à ImageMagick. S'il n'est pas installé le programme le fait juste remarquer et l'image peut être récupéré dans `scenes/`.

Cette option ne prend pas d'argument.

1.3.6 L'option -b (brdf)

Choisis le modèle d'éclairément à utiliser. Voici les différents types d'éclairément possibles :

- 0 : Visualisation simple (sans nuance) utilisé pour un aperçu hyper rapide de la scène ;
- 1 : Modèle de Lambert - éclairément ambiant et diffus ;
- 2 : Modèle de Phong - éclairément spéculaire ;
- 3 : Modèle de Blinn-Phong - amélioration du modèle de Phong.

A noter que le le modèle de Phong et Blinn-Phong utilise tous deux le modèle de Lambert.

Par défaut, le modèle de Blinn-Phong est utilisé.

Exemples d'utilisation : `./kfray -i scene1 -b 2` ; `./kfray -b 0`

1.3.7 L'option -v (view)

Attribue une valeur pour changer l'emplacement de la caméra. Dans la version actuelle l'option ajoute l'argument passé aux coordonnées z de la caméra.

Cette option est utilisée pour le générateur d'images afin de créer des animations.

Valeur à 0 par défaut.

Exemples d'utilisation : `./kfray -i scene2 -v 200.0` ; `./kfray -v -2000.0`

1.3.8 L'option -t (texture)

Désactive les textures procédurales (damier, bruit de Perlin dont bump mapping). Accélère nettement le rendu quand cette option est désactivée.

Activé par défaut.

Cette option ne prend pas d'argument.

1.3.9 L'option -a (anti-aliasing)

Active l'anti-aliasing par méthode de sur-échantillonnage (supersampling). Cette fonction doit être améliorée dans les prochaines versions.

Désactivé par défaut.

Cette option ne prend pas d'argument.

1.3.10 L'option -l (lines) : version parallèle uniquement

Permet de choisir le nombre de lignes de l'image à calculer par chaque processus esclave/ouvrier pour un travail donné.

Imposé à 10 par défaut.

1.4 Utiliser l'interface graphique KF-Ray GUI (“front end”)

Ici peu d'explications sont nécessaires étant donné que l'interface graphique choisit les options à lancer à votre place. Amusez-vous à tester les différentes options et voir ensuite le résultat !

Il vous suffit donc choisir votre fichier de scène, de cocher les bonnes options pour obtenir votre rendu.

Ci-dessous un aperçu des éléments de l'interface.

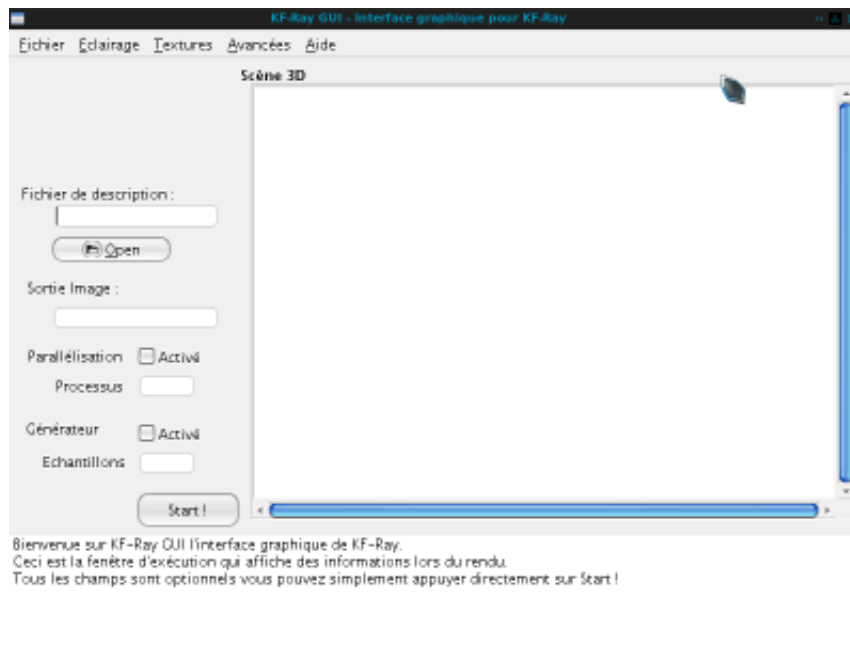


FIG. 1.2 – Interface graphique GUI

Si vous appuyez sur Start! sans rien avoir touché au préalable, vous devriez obtenir la même image que la première.

Il se peut que l'interface graphique possède quelques bugs sur certaines distributions ou configurations de Linux. Merci de le reporter.

Chapitre 2

Créer des scènes personnalisées

2.1 Conseils préalables

2.1.1 L'algorithme de ray tracing

Afin de réaliser au mieux votre fichier de description de scène, il est recommandé d'avoir des notions élémentaires sur la méthode du lancer de rayon (raytracing). Vous pouvez vous référer à notre rapport ou regarder sur Internet, les sites didactiques ne manquent pas.

2.1.2 Conseils sur les coefficients à utiliser

Les valeurs, l'échelle des coefficients sont spécifiques à KF-Ray. Afin d'obtenir l'image de vos rêves, il va falloir vous familiariser avec l'outil et accepter les effets rendus (à moins de changer le code source). Si vous avez une idée très précise de ce que vous voulez, le mieux consiste à tester, changer en tâtonnant plusieurs coefficients jusqu'à obtenir le rendu souhaité.

N'hésitez pas si vous voulez aller plus loin, à vous référer à la documentation du programme afin de mieux comprendre le fonctionnement du raytracer sur <http://kfray.free.fr/doxygen/>. Vous pouvez même modifier dans le code source, quelques formules pour que votre fichier de description possède des coefficients à votre goût.

2.2 Le fichier de description KFR

Avec KF-Ray, il est possible de décrire votre scène 3D. Dans la version actuelle, il est possible de placer des sources de lumières, des sphères et des plans. Vous pouvez personnaliser le matériau constitué par ces deux derniers. Ainsi, vous pouvez attribuer couleurs, textures, coefficients de réflexion, réfraction, spéculaire ainsi que divers effets...

Pour cela, il faut vous munir d'un éditeur de texte et écrire les fichiers au format .kfr dans le dossier scenes/. Il est conseillé de prendre exemple sur les fichiers sceneX.kfr déjà existant dans le répertoire.

Etant donné que KF-Ray utilise son propre interpréteur de fichier (parser), la lecture du fichier n'est pas encore très flexible. Il est conseillé de garder la même syntaxe que dans les exemples proposés.

2.2.1 Règles générales

La syntaxe est très simple. Les cinq objets principaux d'un fichier KFR sont : une scène (scene), des sphères (sphere), des plans (plane), des matériels (material) et des sources de lumière (light). Il est conseillé d'écrire tout en minuscule, vous pouvez cependant, dans la version actuelle, écrire la première lettre de chaque identifiant en majuscule.

Ces objets possèdent des délimiteurs sous forme d'accolades. Les commentaires classiques en C // ... et /* [...] */ sont possibles et non interprétés par KF-Ray.

```
Scène{
  // On précise la scène ici ...
}
```

Chaque objet possède plusieurs caractéristiques qui eux mêmes possèdent des valeurs/attributs.

Chaque caractéristique et les attributs qui leur sont associé peut être défini de plusieurs manières. Voici des exemples qui définissent la couleur RVB d'un matériau. Vous pouvez changer les styles d'écritures :

```
rgb 0.0 255.0 255.0 est correct
RGB = 0.0, 255.0, 255.0 est corret.
Rgb := 0.0 ; 255.0 ; 255.0 est correct.
```

Vous avez un exemple complet de fichier KFR dans la sous-section 3.1.1.

2.2.2 Créer la scène

Une scène est définit par :

- la résolution de l'image
- le nombre et sa liste de sphères, de plans, et de sources de lumières
- la position de la caméra et son point de fuite (vers lequel converge notre vision)

Exemple :

```
scene{
  width = 1024
  height = 768
  materials = 7
  planes = 1
  spheres = 6
  lights = 2
  camera = 0.0, 0.0, -600.0
  distview = 1000.0
}
```

2.2.3 Créer une sphère

Une sphère est définie par :

- son centre,
- son rayon,
- son matériau.

Exemple :

```
sphere{
  center = 220.0, 350.0, 160.0
  radius = 180.0
  material = 2
}
```

2.2.4 Créer un plan

Un plan est défini par :

- sa distance par rapport au point d'origine
- un vecteur normal,
- son matériau

Exemple :

```
plane{
  normal = 0.0, -1.0, 0.05
  distance = 20.0
  material = 7
}
```

2.2.5 Créer un matériel

Un matériel est défini par :

- son numéro d'identification
- son type (normal ou avec texture procédurale en marbre, bois, turbulence...),
- sa couleur en format RVB (chaque flottant est compris entre 0 et 255.0),
- son coefficient de réflexion (compris entre 0.0 et 1.0 exclu),
- son coefficient de réflexion spéculaire (compris entre 0.0 et 2.0),
- son coefficient de rugosité (compris entre 0.0 et 1000.0),
- son coefficient de bruit de Perlin (compris entre 0.0 et 2.0 : 1.0 par défaut),
- son coefficient de bump mapping (compris entre 0.0 et 5.0 : 0.0 par défaut),
- sa densité (compris entre 1.0 et 2.0 : 1.0 par défaut),
- son coefficient de réfraction (compris entre 0.0 et 1.0 : 0.0 par défaut).

À noter que les coefficients qui possèdent des valeurs par défaut sont optionnels et ne doivent pas être obligatoirement être initialisés. KF-Ray s'en charge ce qui facilite l'édition de fichiers.

Exemple :

```
material{
  id = 2
  type = normal
  rgb = 0.0, 0.0, 255.0
  reflection = 0.0
  specular = 0.8
  roughness = 1000.0
  perlin = 1.25
  bump = 0.0
}
```

2.2.6 Créer une source de lumière

Une source lumineuse est défini par :

- le point d'origine de sa source lumineuse,
- son intensité lumineuse.

Exemple :

```
light{  
  position = 800.0, 300.0, 600.0  
  intensity = 0.4, 0.4, 0.4  
}
```

Chapitre 3

Utilisation de KF-Ray basique et avancée

3.1 Exemple de rendu 3D avec KF-Ray

3.1.1 Fichier complet de description de scène

```
Scene{
  Width = 640;
  Height = 480;
  Materials = 3;
  Planes = 0;
  Spheres = 3;
  Lights = 2;
  Camera = 0.0, 0.0, -110.0;
}
//Mat jaune
Material{
  Id = 1;
  Type = turbulence;
  RGB = 255.0, 255.0, 0.0;
  Reflection = 0.5;
  Specular = 1.0;
  Roughness = 60.0;
  Perlin = 1.0;
  Bump = 0.0;
}
//Mat Cyan
Material{
  Id = 2;
  Type = marbel;
  RGB = 0.0, 255.0, 255.0;
  Reflection = 0.5;
  Specular = 1.0;
  Roughness = 60.0;
  Perlin = 1.0;
  Bump = 0.0;
}
//Mat Magenta
Material{
  Id = 3;
  Type = wood;
  RGB = 255.0, 0.0, 255.0;
  Reflection = 0.5;
```



```

    Specular = 1.0;
    Roughness = 60.0;
    Perlin = 1.0;
    Bump = 0.0;
}
Sphere{
    Center = 260.0, 290.0, 0.0;
    Radius = 100.0;
    Material = 1;
}
Sphere{
    Center = 440.0, 290.0, 0.0;
    Radius = 100.0;
    Material = 2;
}
Sphere{
    Center = 350.0, 140.0, 0.0;
    Radius = 100.0;
    Material = 3;
}
Light{
    Position = 640.0, 240.0, -10000.0;
    Intensity = 0.5, 0.5, 0.5;
}
Light{
    Position = 0.0, 240.0, -100.0;
    Intensity = 0.2, 0.2, 0.2;
}

```

3.1.2 En ligne de commandes

Vous venez d'éditer et enregistrer votre fichier de description dans `scenes/mascene.kfr`.

Nous pouvons maintenant rendre votre scène grâce à KF-Ray.

Vous pouvez utiliser simplement la commande : `./kfray -i mascene -d`

Vous pouvez préciser des options et les changer comme bon vous semble :

```
./kfray -i mascene -o monimage -b 2 -t -a -d
```

3.1.3 Avec l'interface graphique

Il suffit de sélectionner votre fichier de scène avec le bouton "Open". Vous devriez apercevoir le texte de votre fichier dans la fenêtre Scène 3D.

Choisissez les options voulus, l'image de sortie (tout est optionnel), et appuyer sur Start!

3.2 Parallélisation avec MPI

Le but du parallélisme avec MPI est de faire travailler plusieurs processeurs/ordinateurs afin de diminuer le temps de calcul et donc générer des images plus rapidement.

3.2.1 Requis

OpenMPI et LAM/MPI sur plusieurs ordinateurs. Vous pouvez utiliser d'autres versions mais nous illustrons nos exemples en utilisant ces bibliothèques.

KF-Ray utilise une méthode d'équilibrage de charge dynamique suivant le modèle Maître/Ouvriers. L'ordinateur où vous allez lancer votre programme sera le processus maître et se chargera de renvoyer et récupérer les travaux de calcul d'images envoyés aux ouvriers.

Seul l'ordinateur maître a besoin du programme KF-Ray. Pour activer la parallélisation il faut compiler KF-Ray avec MPICC au lieu de GCC. Il faut donc taper à la racine du répertoire kfray :

```
./configure CC=mpicc puis make dans le dossier racine de kfray.
```

Vous pouvez maintenant utiliser MPI.

Si vous voulez revenir à GCC tapez simplement :

```
./configure
```

3.2.2 Configurer MPI

Vérifiez que vous avez un accès SSH sur les tous les ordinateurs qui vont être utilisés. Remplissez dans un fichier nommé bhost par exemple l'adresse des différentes machines utilisés. L'adresse de l'ordinateur maître doit être sur la première ligne du fichier.

Exemple :

```
#Fichier bhost réseau local  
127.0.0.1 #PC principal où on lance KF-Ray  
127.0.0.1 #Pour que le maître travaille également  
192.168.0.246 # PC portable de Karin  
192.168.0.240 # PC portable de Florian
```

Enfin voici les étapes pour faire comprendre ce fichier avec LAM/MPI ci-dessous.

Pour vérifier le fichier bhost :

```
recon -v bhost
```

Pour lancer une session LAM :

```
lamboot -v bhost
```

A la fin de l'utilisation de KF-Ray, il faut penser à désactiver LAM :

```
lamwipe -v bhost
```

3.2.3 Rendre une image avec MPI

Voici la commande générale permettant d'utiliser KF-Ray en parallèle sur les différents ordinateurs du bhost.

```
mpirun -v -c [nombre processus] kfray [-options] [arguments]
```

Exemples d'utilisation : `mpirun -v -c 10 kfray -i scene4.kfr -d -l 14`

Vous pouvez également utiliser l'interface graphique en cochant la case "Parallélisme" et en précisant le nombre de processus (vous ne pourrez pas en revanche préciser le nombre de lignes calculés par chaque processus).

3.3 Options avancées

3.3.1 Gagner du temps lors d'une création de scène

Pour un utilisateur souhaitant créer une scène importante sans avoir à perdre du temps lors de la génération de l'image, il est possible d'utiliser l'éclairage simple -b 0 (Menu -> Eclairage -> Sans pour l'interface graphique) pour avoir une vision très rudimentaire de l'image où l'éclairage n'est pas pris en compte on se contente d'envoyer des rayons qui retournent directement la couleur du matériel s'il y a intersection. Il est également possible pour avoir une image un peu plus réaliste de désactiver les textures qui demandent un temps considérable de calcul.

Avec ces options à chaque aperçu de la scène, ajuster les valeurs lorsqu'on édite une scène devient plus agréable.

3.3.2 Anti-aliasing

Afin de lutter contre les phénomènes d'aliasage (dus à un échantillonnage insuffisant de l'image) se manifestant sous les formes suivantes :

- présence de marches d'escaliers sur les contours,
- présence de moirés sur les textures (damier entre autre)
- petits objets entièrement ou partiellement cachés

La fonction antialiasage implémentée dans KF-Ray utilise la méthode de supersampling mais n'est pas encore très convaincante. Vous pouvez toujours l'utiliser le programme essaiera de rendre une image plus précise. Elle sera améliorée dans les versions à venir.

3.4 Créer des animations

3.4.1 Utiliser le générateur d'images avec le GUI

Un des points forts de KF-Ray est de pouvoir générer plusieurs images grâce à l'interface graphique. Cette génération est simple et modifie le point de vue de la caméra suivant le rang de l'image généré. Actuellement, cette version de KF-Ray ne peut que faire des translations simples avec la caméra. Cela reste suffisant pour produire des animations impressionnantes !

Voici comment procéder. Après avoir choisi les différentes options de votre rendu avec l'interface graphique cochez *Générateur activé* et choisissez le nombre d'images à produire (ayez dans l'esprit que 10/20 images est un bon ordre de grandeur). Cliquez sur

Start ! et le programme va commencer à générer une succession d'images sous le nom de anim[rang].png où rang est un nombre à 3 chiffres indiquant le rang de l'image générée.

Une fois les images générées, placez-vous dans le répertoire où les images ont été générés et tapez (nécessite ImageMagick) :

```
animate -delay 10 anim*.ppm
```

Faites attention à ne pas utiliser cette commande si le nombre d'images est trop important. Passez directement à l'encodage vidéo.

```
convert *.ppm *.jpeg
```

3.4.2 Pour créer une vidéo

Vous pouvez installer un encodeur mpeg2enc pour l'utiliser avec ImageMagick et faire un `convert -delay 10 anim*.ppm kfray.mpeg` par exemple. Le problème de cette méthode est qu'elle n'est pas appropriée pour un nombre très important d'images et est peu performante.

Le meilleur choix consiste à utiliser mencoder de MPlayer. Il est par contre indispensable de convertir les images en format JPEG. On pourra utiliser : `convert *.ppm *.jpeg` par exemple (toujours avec ImageMagick).

Note : Une commande utile est `rename` qui permet de renommer un groupe de fichiers.

Une fois les images générées, pour créer une vidéo, dans le répertoire des images jpeg à utiliser, vous pourrez taper :

```
mencoder "mf ://*.jpeg" -mf fps=20 -o kfray.avi -ovc lavc -lavcopts vcodec=msmpeg4v2 :vbitrate=800
```

Reportez-vous au manuel de mencoder pour plus d'informations sur les options. Vous pouvez changer les codecs utilisés, le fps...

Les prochaines versions de KF-Ray...

Voilà vous venez d'apprendre à utiliser les différentes fonctionnalités de KF-Ray. Le programme est toujours en développement donc n'hésitez pas à regarder l'avancement du projet sur <http://kf-ray.googlecode.com>. Vous y avez notamment dans le wiki les listes des futures fonctionnalités de KF-Ray.