Forum: Moteurs de rendu

Topic: [Mini-tuto] Découvrir LuxRender (+GPU)

Subject: [Mini-tuto] Découvrir LuxRender (+GPU)

PostÃ© par: Blob4

Contribution le : 14/11/2010 17:55:03

Salut, je me suis mis à Luxrender et je pense que cela peut être intéressant de rassembler collectivement des informations et nos trouvailles sur ce moteur de rendu, ainsi qu'aider les gens (non anglophones) qui voudraient s'y mettre..

Luxrender est un moteur intéressant à plus d'un titre. Malgré l'avènement de Cycles, encore balbutiant, il reste indétrônable de par sa maturité, ses nombreuses fonctionnalités et la qualité de ses rendus. Luxrender est activement développé et a de nombreuses fonctionnalités dont le rendu en réseau et la gestion interactive du dosage des lumières lors du rendu (via sa GUI externe).

Cerise sur le gâteau, le support OpenCL fait son chemin mais pour le moment sous forme d'hybridation (à la différence de Cycle et SLG2) ce qui permet de préserver les fonctionnalités tout en offrant un gain de puissance avec l'espoir que le photoréalisme devienne encore plus accessible et viable en pratique. Enfin l'exporteur (bien qu'encore en béta) pour Blender 2.6 est de plus abouti.

Voici un mini tuto que je complèterai au fil du temps, grâce à votre aide j'espère, étant donné qu'on a jamais fini de découvrir ce fantastique moteur de rendu.

---

0) Présentation

Luxrender est un moteur de rendu issu de PBRT (Physically Based Ray Tracing), un projet académique, avec pour intention d'offrir en pratique un moteur utilisable par le plus grand nombre. Le concept initial est de modéliser au mieux les lois physiques des phénomènes lumineux, au niveau spectral et non plus dans un simple espace RGB comme Cycles ou SLG2. Il n'y a par ailleurs pas à se soucier comment imiter l'illumination globale (GI) comme sur Blender (avec l'indirect lightning) car celle-ci découle naturellement de l'application des lois de la physique. Les caustiques apparaissent également naturellement, ainsi que bien d'autres "à-cotés" qu'on a tendance à oublier.

Luxrender est pour cette raison dit "non biaisé" car il n'utilise pas de truquage (ce qui bien sûr se paye en temps de rendu plus élevé), mais ce que l'on sait moins est qu'il peut fonctionner aussi en mode biaisé (même si toujours spectral en interne), si on le veut pour des previews ou des effets spéciaux par exemple, le temps requis est alors plus court. Il y a différents algorithmes de rendu disponibles à choisir selon les exigences, certains peuvent parfois mieux convenir à une scène d'intérieur ou une scène d'extérieur, à la présence de caustiques, etc...

Un dernier avantage est de pouvoir prendre en compte les volumes (mais homogène pour le

moment).

1) Installation
L'installation est à cette date relativement facile et consiste en fait à installer un add-on.

D'ailleurs avant de vous lancer, vérifier sur Graphicall.org s'il n'existe pas des builds avec Luxrender tout intégrés (exécutables et scripts). C'est déjà le cas pour la plateforme Win 64bit:
http://www.graphicall.org/lux

Pour les autres:

Première étape: Téléchargez et décompactez l'archive binaire depuis le site.
(Souvenez vous bien où vous décompactez cette archive, vous devrez spécifier l'emplacement plus tard dans Blender pour qu'il puisse trouver les exécutables) :
http://www.luxrender.net/en_GB/download

Si vous voulez un binaire très récent de la branche en cours de développement (quelque soit l'OS), on trouve des weekly builds ici:
http://www.luxrender.net/forum/viewforum.php?f=30&sid=4dfc56261b16cf079a36aa89f8fe9b0d

Si vous tenez à compiler une version vous même (sous nux), il y a un petit tuto très bien pour Ubuntu mais qui peut être utile pour les autres distribs (il explique aussi comment "fetcher" luxblend25):
http://www.luxrender.net/wiki/index.php?title=Building_on_Ubuntu_10.10

Si vous avez récupéré un binaire "stable" depuis le site principal, vous pouvez dors et déjà tester le moteur de rendu, car l'archive contient:
-un répertoire d'exemple avec une scène de test
-parfois un script pour blender 2.4 (à oublier donc)
-pylux pour blender 2.5 (installation optionnelle cf. plus bas)
-3 applications dont:
    -luxrender (la GUI)
    -luxconsole
    -luxmerge

Faisons un petit test pour vérifier qu'il n'y a pas de problème de compatibilité avec le système (libs manquantes, etc..). Lancez l'appli "luxrender" depuis une console, une GUI apparait (ou un message d'erreur qui vous sera utile pour demander de l'aide, souvent c'est que vous n'avez pas téléchargée la bonne archive), ouvrez le fichier d'exemple contenu dans le répertoire et lancez le rendu. Vous devez voir une ville de nuit, éclairée par des lampadaires.
L'image est assez sombre. C'est l'occasion d'essayer les réglages chromatiques, attention la mise à jour peut prendre un peu de temps. Voici une capture de l'interface, avec le fichier d'exemple et des réglages "maison" pour rendre l'image moins sombre:

Deuxième étape: Installer l'add-on LuxBlend2.5
On trouve le fichier ici:
http://src.luxrender.net/luxblend25
Il faut cliquer sur zip ou bzip ou le format de son choix en bleu en haut. Décompactez l'archive et copiez le répertoire "luxrender" là où est installé blender 2.6.x, plus précisément dans "scripts/addons". (Attention sous Windows, le répertoire "script" est un fichier caché ".scripts" qui peut se trouver à un emplacement différent).

(La page web en anglais pour plus d'informations).

Voilà le principal est fait, donc plutôt simple. Une fois qu'on a lancé Blender, dans "User preferences", cliquez Add-ons et dans la catégorie Render on voit l'option Luxrender qu'il faut cocher. N'oubliez pas de sauver vos préférences (Ctrl+U).

Troisième étape (optionnelle): Pylux
Copiez le fichier pylux qu'on trouve dans l'archive de Luxrender dans le répertoire "luxrender" qui vient d'être copié dans "scripts/addons".

Pylux sert à plusieurs choses, la première est d'autoriser la preview en temps réel des matériaux comme fait Blender avec l'internal, la deuxième est de permettre la totale intégration du moteur en interne depuis Blender pour les rendus (on aura pas l'impression qu'un moteur externe est utilisé car tout se fera depuis Blender).

2) Premiers pas et configuration

Quand on a activé l'add-on, on doit voir en haut (dans la scène principale du cube par exemple) dans le menu déroulant Luxrender disponible en plus des autres moteurs. Il suffit de le choisir et on verra que les panneaux sur la droite changeront.

La chose la plus importante à faire à présent est de choisir la façon dont Blender va appeler LR. A l'écran des rendus (icône appareil photo) aller à l'onglet "Luxrender Engine Configuration", on a le choix dans "Rendering mode", entre Internal ou External. Le premier permet l'intégration totale du moteur, le deuxième se comporte comme simple un exporteur. Il va exporter la scène et accessoirement lancer LR soit en mode GUI indépendante et persistante soit en mode console. Le mode GUI à l'avantage de permettre de faire joujou avec les réglages lumineux interactivement. On ne peut pas faire cela depuis Blender malheureusement.

A noter:
-il faut bien cliquer "Run renderer" en blanc sinon le script se comporte comme un exporteur passif et on doit alors lancer LR en manuel et charger les fichiers exportés.

-si on interrompt la rendu depuis Blender (touche Esc) en mode External le processus LR n'est pas interrompu et continu à tourner en toile de fond. C'est juste que Blender arrêtera de mettre à jour l'image. Il faudra donc fermer soit même LR.

Dernière chose d'importance et primordiale avant de lancer un premier rendu, il faut bien indiquer le chemin vers le dossier ou vous avez décompacté LR dans "Path to luxrender"

Très important également, LR étant basé sur la physique **il est impératif de travailler en respectant la taille réelle des objets, 1 BlenderUnit = 1 mètre. Et il faut régler Scale à 1 dans Scenes/Units.**

Voilà, à partir de cette étape vous êtes normalement paré.

3) Setups d'exemple

Si on essaye de rendre directement le cube celui ci apparaitra en noir et blanc, ça vient du fait que LR (nous utiliserons cette abréviation pour Luxrender) gère l'éclairage différemment que Blender. Il n'y a pas (encore dans luxblend) de couleur d'environnement, le fond est par défaut toujours noir (cependant on peut utiliser une source type "Sun" qui a un réglage générant un ciel, ou "Hemi" avec une image HDRI en fond).

Pour les faces noires du cube, comme il n'y a aucun corps réverbérant la lumière, celles ci apparaissent noires comme cela serait le cas dans l'espace par exemple. Il suffit d'ajouter un plan en dessous pour les faire ressortir. Bien que supportée la lampe par défaut est déconseillée (dixit la doc un simple point lumineux étant une singularité non présente dans la réalité) et il faut utiliser plutôt des surfaces comme les Areas. Autre particularité on peut rendre un mesh lumineux et l'utiliser comme lampe. Cependant plus le mesh a de face, plus les calculs seront longs.

Il y a quelques exemples sur le wiki pour démarrer bien qu'ils se réfèrent encore a blender2.4
http://www.luxrender.net/wiki/index.php?title=Creating_a_soap_material

Sinon pour importer une image OpenEXR et obtenir un éclairage depuis un fond HDRI il faut choisir "Hemi" comme type de lampe, une ligne "HDRI map" apparait qui permet de choisir une image. Le type de fichier peut être soit "angular" (sphère) ou latlong (pour les images longitudinales en 180x360°), le format jpeg marche mais le .exr est recommandé.

(à compléter)
J'essayerai de poster quelques blends d'exemple..

4) Les différents modes de rendu

A l'onglet "Surface Integrator" se trouve le réglage le plus stratégique car il permet de sélectionner le type de rendu que l'on souhaite, ce qui va influer bien-sûr sur les temps de calculs, de nombreuses options supplémentaires peuvent ensuite apparaitre.

Il faut différencier en premier lieu les algos biaisés et ceux non biaisés.

Ex-Photon Map: Biaisé. Il s'agit de la technique classique du photon mapping (je crois utilisé par Yafaray). Le "Ex" était ici pour "expérimental", mais à ce jour cet algo n'est plus considéré comme tel. (ceci dit, chez moi j'ai eu quelques plantages dans les tests, peut-être un pbm de RAM insuffisante). Ce moteur convient bien aussi pour les animations, mieux que le "distributed path" selon certains.

Instant Global Illumination: Un algo issu de PBRT qui est pour l'instant biaisé car il n'implémente pas les derniers développements de PBRT2. Cet algo se base sur un ensemble de lumières virtuelles générées lors de l'initialisation de la scène qui permettent de simuler l'illumination globale.

Distributed Path: Non biaisé. C'est un algo dérivé du classique algo de Path tracing. A l'origine cet algo est non biaisé mais la doc stipule qu'on peut grandement le régler pour le rendre biaisé. L'avantage étant de permettre un bon compromis temps/qualité pour les animations. C'est celui que j'ai le plus utilisé.

Direct Ligthning: Biaisé, c'est une sorte d'algo de test pour les preview, aucun caustique ou illumination globale n'apparaitra. Logiquement c'est l'algo le plus rapide, rendu type ratraycer primitif donc, aspect très synthétique.

Path: Non biaisé, algo classique de path tracing. (et aussi le seul algo permettant l'OpenCL pour l'instant).

Bi-directionnal: Non biaisé, algo de path tracing plus évolué car le chemin de la lumière est pris en considération dans les deux sens, depuis la caméra et la lumière, ce qui permet des optimisations. C'est je crois l'algo à utiliser en préférence (c'est d'ailleurs l'algo par défaut). Note: les caustiques se formeront plus vite avec cet algo.

SPPM: Stochastic Progressive Photon Mapping. Cet algo est en cours d'implémentation et ne sera disponible que sur les prochaines version de LR. Il deviendra probablement l'algo par défaut, il a les avantages du photon mapping sans les inconvénients.

5) Petites choses à savoir

-Par défaut il n'y a pas de fin à un rendu et LR va tourner indéfiniment jusqu'à ce que vous le stoppiez. Les images sont constituées par la somme des échantillons (samples) calculés. Plus on en obtient, plus l'image est nette. On peut cependant cocher et spécifier dans Blender le paramètre "SPP halt" qui correspond au nombre de samples par pixel auquel le rendu sera arrêté (très utile dans le cadre des animations).

-Par défaut avec le tonemappeur Reinhard, LR s'adapte au niveau d'éclairage comme la pupille de l'oeil humain. Ainsi changer la puissance d'une lampe si elle unique dans la scène ne changera rien au niveau du rendu, car LR compensera pour afficher une scène avec toujours la même luminosité. Cela peut être changé toutefois, pour les animations par exemple, en choisissant un autre type de tonemappeur (cf. plus bas).

-Les algo Ex Photon Map et Instant Global Illumination ne supportent pas les light groups.

-Il existe un certains nombres de différences ou limitations entre l'internal et LR (pas de SSS mais certains matériaux permettent de l'imiter), le script luxblend25 n'est pas encore finalisé (certaines particules et les hairs ne fonctionnent pas, les NURBS ne sont pas supportés,...).

-pour utiliser un PC en mode serveur tapez "luxconsole -s" dans un terminal. Il se met alors en attente et pour utiliser cette ressource il suffit de rentrer dans Blender l'IP de cette machine à l'onglet (option chochée) "Use Networking".

-"luxmerge" permet de fusionner des rendus dispersés sur différents ordinateurs. Les "films" sont fusionnés pour donner une image finale moins bruitée.

-on peut interrompre, quitter LR et reprendre un rendu plus tard, et ce à tout moment à condition de sauver le "film" (fichier .flm), et de le restorer par la suite.

-Les scènes exportées sont décrites dans un langage dont on peut connaitre les spécifications ici:
http://www.luxrender.net/wiki/index.php?title=Scene_file_format

-il existe plusieurs projets de fermes de rendu free ou commerciaux: vswarm, une initiative utilisant Boinc en développement, un projet commercial greenbutton:
http://support.vswarm.com/wiki/AppLuxrender
http://www.luxrenderfarm.de/luxfarm/
http://www.greenbutton.net/apps/luxrender

-le lien du wiki pour la doc:
http://www.luxrender.net/wiki/index.php?title=Main_Page

-une vidéo en anglais principalement sur l'installation de luxblend2.5
http://www.blendercookie.com/2010/11/11/introduction-to-luxrender-in-blender-2-5/

6) Utiliser le GPU et OpenCL

Quelques petites conditions sont nécessaires, la première pour utiliser OpenCL il faut une version >= 0.8 compilée AVEC le support OpenCL. Vous devez impérativement avoir installé "pylux", sinon vous ne verrez pas les options pour sélectionner le mode GPU. Vous aurez peut-être en plus à installer les derniers drivers ATI ou Nvidia.

Pour les cartes ATI, il faut installer le ATI Stream SDK:
http://developer.amd.com/gpu/ATIStreamSDK/downloads/Pages/default.aspx

Ensuite dans l'onglet "LuxRender Engine Configuration" à Renderer choisir le mode Hybrid:

Enfin à l'heure où nous écrivons ces lignes, seul l'algo "Path" peut exploiter le GPU, donc il convient de le sélectionner à l'onglet "Surface Integrator", ensuite cliquez "Advanced" et mettez "Light strategy" à "One".

Dans les dernières builds de LR, il y a une jauge GPU qui permet de connaître la part de la carte graphique dans le rendu.

David Bucciarelli est le principal contributeur du support OpenCL, ce travail a été d'abord incarné sous une forme experimental de raytracer appelé "SLG". Mais l'effort porte à présent sur le support OpenCL au sein même de LR v0.8.

Dans le futur différents modes seront disponibles, soit on pourra travailler en Hybride, c'est à dire CPU+GPU, soit en full GPU.

Un topo sur OpenCL et LR:
http://www.luxrender.net/wiki/index.php?title=Luxrender_and_OpenCL