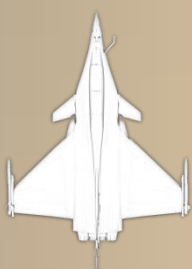
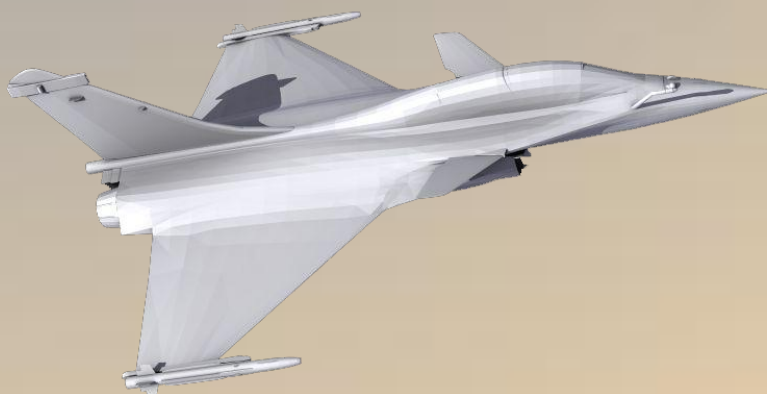
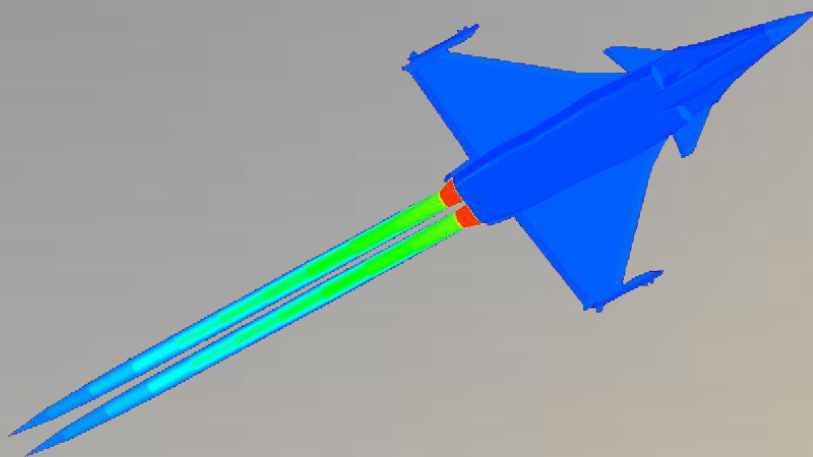


Manuel Utilisation

Advanced Rendering Tools

Réalisation d'une solution de rendu graphique sous GPU



Master Physique et Informatique

Emmanuel Hermellin



Informations complémentaires sur le manuel d'utilisation:

Si vous voulez utiliser ou vous faire une idée des performances de ARTGpu, vous pouvez l'installer sur votre machine la version utilisateur (il faut une carte Nvidia compatible Cuda (de type Fermi minimum), nécessite MinGW) ou la version développeur (il faut une carte Nvidia compatible Cuda (architecture Fermi), avoir installé Cuda Toolkit et Optix SDK ainsi que Visual Studio Express, Cmake et enfin MinGW).

Lien de téléchargement ARTGpu version utilisateur :

Lien de téléchargement ARTGpu version développeur :

J'ai aussi mis un site internet présentant le stage et tout le travail accompli pendant ces quelques mois. Vous trouverez sur ce site tous les documents, fichiers d'installations, rapports et manuels, photos et présentations du stage...

Lien :

Manuel d'utilisation ARTGpu

a. Installation version développeur

Pour installer le logiciel, il faut suivre plusieurs étapes. Il faut au préalable avoir installé Visual Studio (2008 de préférence), le toolkit CUDA, le SDK OPTIX et MinGW et Cmake.

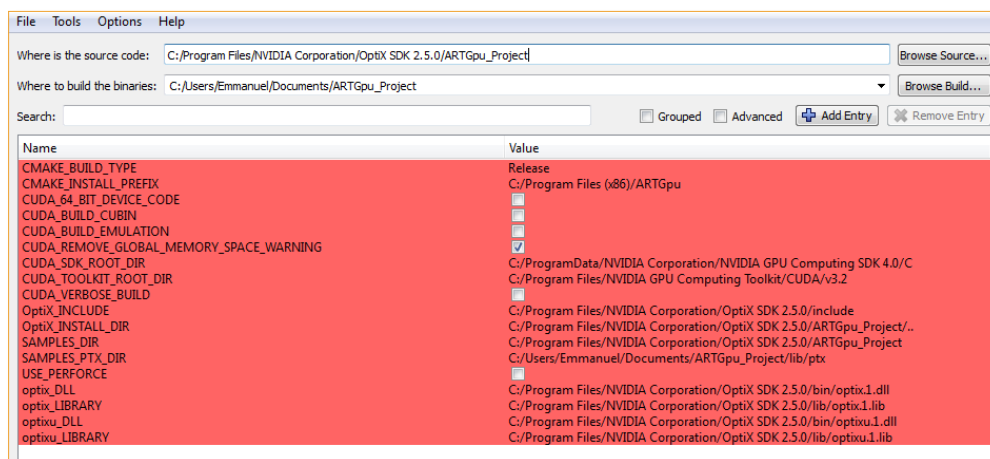
Télécharger Cmake: <http://www.cmake.org/cmake/resources/software.html>

Copier le dossier *ART_GPU* dans le dossier par défaut d'installation d'Optix:

C:\Program Files\NVIDIA Corporation\OptiX SDK 2.5.0

Lancez l'application *cmake-gui* et définir le chemin de vos sources (*C:\Program Files\NVIDIA Corporation\OptiX SDK 2.5.0\ARTGpu_Project*) et le dossier destination qui contiendra tous les fichiers de compilations, exécutables,... (par exemple *C:\Users\X\Documents\ARTGpu_Project*).

Cliquez sur **configurer** et choisir dans la nouvelle fenêtre votre IDE (exemple : Visual Studio 2008). **ATTENTION** bien choisir x64 si l'IDE fonctionne en 64bits et x32 s'il fonctionne en 32 bits. Et cliquer **OK**. Cmake va mouliner et des informations vont apparaître sur votre écran.



Fenêtre de configuration Cmake.

Cliquez une nouvelle fois sur **configurer** puis sur **générer**. ARTGpu est prêt à être compilé.

Rendez-vous dans le dossier destination que vous avez défini précédemment et ouvrez le fichier .sln qui vous ouvrira VS2008. Faites un build du projet: **Build/Build Solution**. Maintenant que le build est terminé, les exécutables de ARTGpu sont disponibles.

Pour finir il faut modifier le path dans le script perl *Start.pl* :

my \$chemin = "C:/Users/xxxxxxxxxx/Documents/ARTGpu_Final/bin/Debug/"; Remplacer le chemin par le vôtre.

Les scripts perl et l'exécutable *nconvert.exe* doivent être dans le répertoire **Debug** de la solution donc dans le même répertoire que les exécutables. Et ensuite lancer le script (vous devez utiliser Mingw): *perl Start.pl*

Voilà ARTGpu est utilisable !

b. Installation version utilisateur

Récupérez le fichier *ARTGpu-setpu.exe* et installez-le comme n'importe quel logiciel windows. La version utilisateur ne permet pas de modifier le code source. On ne peut pas non plus choisir le répertoire d'installation pour des raisons de fonctionnement dans le code source.

c. Fonctionnement détaillé de ARTGpu

Lancement du script perl grâce à MinGW : *perl Start.pl* (ou grâce au raccourci : version utilisateur)

```

      000  000  00000  000  000  0  0
      0  0  0  0  0  0  0  0  0  0
      00000 0000  0  000 0 000 0000 0 0
      0  0  0  0  0  0  0  0  0
      0  0  0  0  0  0  000  0  000

26/6/2012 10:54:13
dota-vinci

Menu Principal
-----
1- Choisir une géométrie 3D .obj
2- Lancement ARTGpu - Outils de rendu
3- Information système / Compatibilité
4- About
5- Quitter

Fichier de géométrie choisie:
Choix: _

```

Vous vous trouvez dans le menu principal de ARTGpu. Vous avez donc plusieurs choix : entrer le chemin vers le fichier obj, lancer le rendu de la géométrie (il faut bien sûr en avoir rentrée une avant), tester la compatibilité de son poste de travail, afficher les informations de ARTGpu et enfin quitter.

Si on veut effectuer un rendu, la première chose à faire est de rentrer un fichier .obj (commande 1 dans le menu). Entrez le chemin vers le fichier obj (en utilisant la notation linux: c:/.../.../...) puis *Entrée*.

```

Choix: 1
Veuillez entrer le chemin vers le fichier obj:

```

Après avoir entré la géométrie, vous verrez votre choix s'afficher dans le menu :

```

Menu Principal
-----
1- Choisir une géométrie 3D .obj
2- Lancement ARTGpu - Outils de rendu
3- Information système / Compatibilité
4- About
5- Quitter

Fichier de géométrie choisie: c:/ObjFiles/rafale.obj
Choix: _

```

Vous pouvez maintenant lancer un rendu avec la géométrie sélectionnée, en utilisant la commande 2 du menu.

```

      000  000  00000  000  000  0  0
      0  0  0  0  0  0  0  0  0  0
      00000 0000  0  000 0 000 0000 0 0
      0  0  0  0  0  0  0  0  0
      0  0  0  0  0  0  000  0  000

Bienvenue dans ARTGpu v1.0 Juin 2012
Logiciel de démonstration technique: rendu graphique

```

Choisir le module de rendu que l'on veut utiliser : raytracing, photon mapping ou path tracing.

```

!Choix du module de rendu:
-----
!m Path Tracing MCMC
!r Raytracing
!p PhotonMapping
!=> _

```

Vous avez maintenant trois choix (ces choix sont les mêmes pour les trois modules de rendu) :

- ⇒ **miss**: utilisation d'une couleur d'arrière-plan unie que vous rentrerez vous même.
- ⇒ **env**: utilisation d'une image type hdr qui fera office d'environnement map de la scène. Le programme vous demandera de rentrer son chemin juste après et si une image est déjà présente il vous demande si vous voulez l'utiliser ou non.
- ⇒ : juste appuyer sur **Entrée** sans choisir d'option chargera les paramètres par défaut.

```
!Background options:
!-----
! miss      Couleur d'arrière plan uniforme
! env       Environnement Mapping
! =>
```

Viens ensuite le script matériaux.pl qui s'exécute et qui va soit vous afficher les informations du mtl existant et faire les conversions des textures si nécessaire, soit créer un fichier mtl grâce à un fichier référence Data_BRDF fournit par Crira.

```
!Gestion des fichiers materiaux:
!-----
!Le fichier rafale.mtl existe!
!Aucune conversion d'image nécessaire!
!Il y a 3 materiaux dans le fichier.
```

Un choix vous est ensuite demandé (si vous avez choisi le module Raytracing) : Utiliser les jets ou pas ! Si vous choisissez de les utiliser, le script jets.pl va convertir le fichier jet_SDM (format bdd) fournit par Crira au format obj. Si vous choisissez dans les options de rendu l'option jets <n> (avec le nombre de jets à afficher, ARTGpu va intégrer les jets dans la scène).

```
!Gestion des jets:
!-----
!Voulez vous générer les fichiers de jets? <y/n>
! =>
```

Ensuite vous avez deux types d'options : les options **OpenGL** qui touchent la fenêtre windows et les options de rendu qui vont modifier la façon dont ARTGpu fait les calculs.

```
-----
ARTGpu      Module Raytracing
-----

!Gestion des jets:
!-----
!Voulez vous générer les fichiers de jets? <y/n>
! => n
!Gestion de jets non choisi!

!OpenGL options:
!-----
! -d | -d=<width>x<height> Set image dimensions
! -D | -D=<num_devices>   Set desired number of GPUs

!ARTGpu options:
!-----
! ao      Use progressive Ambient Occlusion shader
! aop     Use progressive Ambient Occlusion and Phong shader
! m | m <n> Multiplier for the number of AO samples
! r | r <scale> Scale ambient occlusion radius
! ns      Use normal shader <show illumination map>
! ds      Use one bounce Diffuse shader
! fr      Use Fresnel shader <fresnel reflexion>
! ph      Use Phong shader <ambient, shadows, reflexion>
! aa      Use subpixel jittering to perform antialiasing
! oc      Use orthographic camera <cannot use AO mode with ortho>
! l | l <scale> Scale lights by constant factor
! kd      Kd Tree acceleration
! lbvh    lbvh acceleration
! bvh     bvh acceleration
! t | t <n> Progressive Render Time <par default 10s>
! axe     Display x y z axes
! jets    Utilisation du fichier jet pour l'avion

Paramètres <chacun séparé par un espace>=> _
```

Pour rentrer une option, vous tapez juste la commande qui vous convient (exemple: aop pour ambient occlusion and phong). Vous pouvez taper plusieurs commandes à la suite du moment qu'elles sont séparées par un espace.

Il faut comprendre que tous ces paramètres ainsi que ceux tapés précédemment sont passés à ARTGpu par l'intermédiaire des tableaux d'arguments argv / argc. Exemple de ligne de paramètres: *aop aa m 4 t 20*

ARTGpu se lance maintenant que la phase de préparation et de paramétrage est finit (sauf si vous avez choisi *miss*, ARTGpu vous demandera la couleur (format float RGB) d'arrière-plan). Si vous avez choisi le module photon mapping, ARTGpu vous demandera les paramètres de la lumière à utiliser. Le premier écran présente les options propres à la fenêtre windows qui vient de s'ouvrir avec les interactions souris / clavier possibles.

```

----- Lancement de ARTGpu -----
Mouse interaction:
left mouse      Camera Rotate/Orbit
middle mouse    Camera Pan/Truck
right mouse     Camera Dolly
right mouse + shift Camera FOU

Keyboard interaction:
q Quit
f Toggle full screen
b Start/stop a benchmark
d Toggle frame rate display
s Save a frame to 'SavePicture.ppm'
m Toggle memory usage printing
c Print camera pose
l Print Sun pose (x X ! y Y ! z Z)

Aucun parametre de rendu entre!
Aucun parametre d'environnement entre!

```



Le deuxième écran est là pour informations en donnant le temps de chargement, compilation, création de l'arbre...

```

Time to load geometry: 0.86546 s.
Time to compile kernel: 0.690303 s.
Time to build AS      : 1.57013 s.

```

En quittant la fenêtre OpenGL, ARTGpu se ferme et va automatiquement vérifier si vous avez effectué des sauvegardes d'images et les convertir / déplacer si il en trouve.

```

----- Conversion des images -----
Nombre d'images: 0

```

Pour finir, ARTGpu vous propose de relancer le calcul.

```

----- Fermeture de ARTGpu -----
Voulez vous relancer un calcul (y/n): y

```

Si vous choisissez non, ARTGpu reviendra au menu principal.

Intéressons-nous maintenant au module information et compatibilité (choix 3 dans le menu):

```

ARTGpu      Informations Systèmes
-----
#----- Environnement et Cpu -----
|
| Poste de travail:      DOTA-VINCI
| Systeme d'exploitation: Microsoft Windows 7 Professionnel
| Type du systeme:      x64-based PC
| Nombre(s) Processeur(s): 1
|   Détails: Intel64 Family 6 Model 26 Stepping 5 GenuineIntel ~3193 MHz
| Memoire RAM totale:    16 374 Mo
| Memoire RAM disponible: 13 755 Mo
|

```

...

```

#----- Compatibilite -----
|
| Votre Ordinateur est compatible avec ARTGpu
|

```

d. Historique des versions

