TP n° 8 : Introduction à CUDA

0. Introduction

CUDA est l'environnement de programmation de NVIDIA pour la programmation des GPU.
L'objectif de ce TP est de manipuler des codes source et de mesurer les temps d'exécution sur CPU et sur GPU. On utilise des programmes « exemples » de NVIDIA, auxquels sont parfois ajoutées de mesures de temps supplémentaires.

Environnement CUDA

On utilise Visual Studio10 sous Windows7. La procédure est la suivante :

- Les projets vs2010 sont accessibles via Programmes/NVIDIA Corporation/CUDA Samples/v5.0. Produit scalaire est accessible sous la rubrique Advanced. La convolution séparable et dct8x8 sont accessibles sous la rubrique Imaging.
- Cliquer sur le projet vs2010 correspondant au programme pour lancer Visual Studio.
- Utiliser les options « realise » et « x64 ».
- Commande « Build solution » ou « Rebuild solution »
- L'exécution est lancée par la commande « Start without debugging (Ctrl+F5).

1. Produit scalaire

Le code est fourni dans les exemples de NVIDIA, avec des variantes disponibles sur le site du cours. Voir également le document sur le site du cours
Exécuter le programme « produit scalaire » et mesurer les temps d'exécution en distinguant les temps de calcul CPU et GPU et les temps globaux en incluant pour le GPU les temps de transferts entre CPU et GPU.

- Exécuter d'abord la version NVIDIA
- Remplacer certains programmes par ceux fournis sur le site du M1 (pour plus de prises de temps)

2. Convolution séparable.

Le code est fourni dans les exemples de NVIDIA, avec des variantes disponibles sur le site du cours. Voir également le document sur le site du cours

Mesure des temps d'exécution CPU et GPU

- Exécuter d'abord la version NVIDIA
- Remplacer certains programmes par ceux fournis sur le site du M1 (pour plus de prises de temps)

Exploration de l'espace des configurations.

La configuration 16x8 est une configuration de bloc de threads très efficace pour de nombreux problèmes.

1. Essayer d'autres configurations de blocs de threads pour la **convolutionRowsGPU**
2. Essayer d'autres configurations de blocs de threads pour la **convolutionColumsGPU**
3. Indiquer toutes les configurations trouvées plus rapide que la configuration initiale (16x8 et 16x8).

3. DCT 8x8

(Optionnel) – Même travail.