

# TP5

## **Objectifs :**

- Traitement d'image (partie 2)

## **Ressources**

- <http://docs.nvidia.com/cuda/index.html>
- [https://docs.nvidia.com/cuda/pdf/CUDA\\_C\\_Programming\\_Guide.pdf](https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf)

## **Setup**

- haswell-cuda ou 192.168.80.64
- Login : prenom.nom (minuscule)
- Passwd : cuda (à changer)
- Vérification bashrc, PATH et LD\_LIBRARY\_PATH

**Git :** <https://github.com/hpc-apps/Cuda.git>

## **Exercice 1**

- Récupérer le programme `sobel.cpp`
- En vous inspirant du TP4, créer deux kernels CUDA distincts pour la mise en œuvre des algorithmes grayscale et de Sobel .

## **Exercice 2**

- Fusionner les kernels grayscale et Sobel afin d'implémenter un unique kernel CUDA pour ces deux algorithmes.

## **Exercice 3**

- Instrumenter les codes CPU et GPU et comparer les temps d'exécution pour les 2 tailles d'image.

## **Exercice 4**

- Analyser l'accès aux données et modifier l'implémentation de l'algorithme de Sobel afin d'utiliser la mémoire de type *shared* (version avec deux kernels CUDA distincts puis version fusionnée)
- Anticiper les recouvrements nécessaires pour le partage de données entre blocs et les synchronisations

## **Exercice 5**

- Modifier l'implémentation précédente afin d'utiliser des Streams et optimiser le recouvrement calcul/transfert de données.
- Instrumenter cette nouvelle version du code et comparer les performances.