

# Transposition GPU d'un algorithme de traitement d'images type 'snake'.

Gilles Perrot<sup>1</sup>, Stéphane Domas<sup>1</sup>, Raphaël Couturier<sup>1</sup>, Nicolas Bertaux<sup>2</sup>

<sup>1</sup>Institut FEMTO-ST

Rue Engel Gros, 90000 Belfort, France.

prenom.nom@univ-fcomte.fr

<sup>2</sup> Institut Fresnel, CNRS, Aix-Marseille Université, Ecole Centrale Marseille,  
Campus de Saint-Jérôme, 13013 Marseille, France.

nicolas.bertaux@ec-marseille.fr

## RÉSUMÉ

En traitement d'images, une limite à l'utilisation d'un grand nombre d'algorithmes avancés demeure le temps de calcul de leurs implémentations. Parallèlement, les gains obtenus aujourd'hui en puissance de calcul le sont essentiellement grâce l'emploi d'architectures parallèles de diverses natures. Parmi ces solutions, les GPGPUs (processeurs graphiques à usage général) sont une réponse économique et performante pour l'implémentation de certaines classes d'algorithmes. Toutefois, leur architecture très spécifique ainsi que l'interaction avec le processeur hôte ne permettent pas de garantir, dans tous les cas, une transposition simple et efficace. À tel point qu'il n'est pas rare de voir des implémentations GPU moins performantes que l'implémentation CPU de référence. Nous proposons d'illustrer ces aspects en s'appuyant sur l'exemple d'un algorithme de segmentation par contour actif, orienté régions (snake). Après avoir rappelé brièvement le principe du *snake* étudié, nous détaillerons la méthode employée pour paralléliser cet algorithme sur GPU, en nous focalisant sur certains aspects génériques et symptomatiques. Nous évoquerons en particulier, en donnant autant que possible des extraits de code :

- la représentation des données. Elle est une des clés de la transposition des algorithmes et est étroitement liée au niveau de parallélisme appliqué.
- la génération d'images intégrales. Elle représente un type de prétraitement très employé.
- les étapes de réduction. Par exemple les sommes, tris ou recherches d'extrema. Elles sont un point délicat car les GPUs ne sont clairement pas conçus pour que ces traitements y soient exécutés de manière optimale. Un certain nombre de travaux apportent des réponses que l'on présentera.
- l'optimisation de la grille de calcul. Contrairement à ce que préconise le principal fabricant de GPU et ainsi que l'a montré Volkov, la maximisation de l'occupation (occupancy) ne permet pas toujours de tirer le meilleur parti de la puissance de calcul du GPU. De ce point de vue, les architectures récentes peuvent représenter un recul.

Chacun des points abordés sera également l'occasion de présenter les spécificités et les contraintes des accès aux différentes zones mémoire des GPU (globale, partagée, textures, registres).