

Rapport sur le mémoire de thèse intitulé

“Algorithmes rapides pour le filtrage et la segmentation des images bruitées sur GPU”

présenté par

M. Gilles PERROT

Beaucoup d'applications du traitement d'images numériques ont besoin d'être très rapides, voire « temps réel », c'est-à-dire nécessitant un temps de calcul perçu à l'échelle humaine comme quasi instantané. Or les nombreux calculs engendrés par ces traitements ne permettent souvent pas d'atteindre de telles vitesses, limitant ainsi les champs d'application. Grâce à l'avènement des processeurs graphiques massivement parallèles et très bon marché (appelés GPU, pour Graphics Processing Unit), beaucoup de traitements se sont vu considérablement accélérés, au point même d'avoir introduit une terminologie spécifique : le GPGPU, pour « General Purpose processing on Graphics Processing Units ». Malheureusement, il ne suffit pas de transcrire un algorithme séquentiel CPU sur GPU pour garantir un gain notable en performance. Aussi, l'écriture d'un algorithme destiné à être exécuté sur GPU, ce dernier correspondant à une architecture massivement parallèle de type SIMD (Single Instruction Multiple Data), peut s'avérer être un travail difficile et délicat. Il est souvent nécessaire de remanier l'ensemble de l'algorithme ainsi que les structures de données qui lui sont associées. M. Perrot propose de s'attaquer précisément à ce problème. En pratique, il s'agit de porter un ensemble d'algorithmes de segmentation et de réduction de bruit dans des images numériques sur GPU afin d'obtenir des accélérations de traitements qui soient – dans la mesure du possible – importants (au moins d'un ordre de grandeur).

Organisation du manuscrit

Le manuscrit fait 130 pages. Il est structuré en quatre parties, la première correspondant à l'introduction et la dernière au chapitre de conclusion. La seconde partie présente un état de l'art du traitement des images numériques. Elle est composée de trois chapitres. Cet état de l'art se concentre essentiellement sur les notions d'image et de bruit. Après une présentation rapide au chapitre 2 de l'architecture des cartes graphiques de marque NVidia ainsi que les enjeux de la programmation sur ce type de cartes, le chapitre 3 commence par décrire les quatre principaux modèles de bruit dans une image : le bruit Gaussien, le speckle, le bruit « sel et poivre » et le bruit de Poisson. Ce chapitre est très succinct (trois pages), mais offre un très bon aperçu de cette notion en traitement d'images. Le chapitre suivant propose alors d'approfondir la problématique de la réduction du bruit dans les images. Pour chacun des filtres décrit (convolution, médian, bilatéral, etc.), M. Perrot analyse les implémentations sur GPU qui ont été d'ores et déjà proposées. De même le chapitre suivant décrit un ensemble de techniques de segmentation par régions et par contours actifs, puis analyse les implémentations sur GPU proposées. L'ensemble de ces chapitres introduit très bien le contexte et les enjeux. Ils sont agréables à lire et se concentrent sur l'essentiel.

300 bd Sébastien Brant
BP 10413
F-67412 Illkirch Cedex
Tél. : (33) 03 68 85 45 59
Fax : (33) 03 68 85 44 55

<http://icube.unistra.fr>

La troisième partie, composée de quatre chapitres, présente les contributions de M. Perrot. Je détaille ces contributions dans ce qui suit.

Contributions

La première contribution de M. Perrot, décrite au chapitre 6, consiste en une implémentation sur GPU de l'algorithme de segmentation par contours actif de type « polygonal snake ». La méthode se propose de découper le processus en trois étapes clé. Une première étape (représentant environ 20% des ressources globales) consiste en un pré-calcul des images cumulées. Le calcul est effectué en trois niveaux (donc trois noyaux, appelés kernel) qui ont besoin d'une synchronisation entre eux. Il s'agit essentiellement de cumuler des sommes partielles, ce qui est un calcul facilement parallélisable et généralement bien adapté aux GPUs. La seconde étape mobilisant quasiment le reste des ressources consiste à déplacer les nœuds du contour polygonal discrétisé en pixels. Un déséquilibre de charge peut apparaître en raison des longueurs différentes des segments. M. Perrot propose donc de compléter les blocs de pixels par des valeurs nulles. Par ailleurs, il peut se poser le problème de la validité du contour généré, ceci pouvant potentiellement engendrer des problèmes de convergence de l'algorithme. Pour limiter ce problème, M. Perrot propose d'alterner les traitements entre nœuds d'indices pairs et impairs. Il s'agit là d'une des principales différences entre la version séquentielle existante et la nouvelle version parallèle proposée. Un déséquilibre de charge se produit, car certains processus (threads) peuvent devenir inactifs. Enfin, on remarque que plus les segments sont longs (i.e. moins il y a de sommets pour le contour) plus le parallélisme est efficace car plus l'homogénéité des calculs est grande. La troisième étape consiste à déterminer un « meilleur » contour initial, ce dernier pouvant avoir une influence notable sur la vitesse de convergence de l'algorithme. Pour cela les capacités du GPU sont également exploitées : il s'agit essentiellement d'évaluer un grand nombre de contours rectangulaires et d'en « conserver » le meilleur au sens du maximum de vraisemblance. En conclusion de ce chapitre M. Perrot montre qu'un gain de performance de l'ordre de 6 à 15 est possible grâce à l'emploi du GPU.

La seconde contribution porte sur la réduction du bruit par recherche de lignes de niveau. Ce chapitre nécessite pour sa compréhension une bonne connaissance de la notion d' « isoline » dans une image. Les isolines permettent d'effectuer un débruitage en préservant des transitions nettes. Elles sont construites segment par segment (approche « gloutonne ») en utilisant un test de vraisemblance GLR (Generalized Likelihood Ratio), chaque segment pouvant avoir une configuration parmi 32 possibles. Afin d'éviter de parcourir toutes les configurations possibles, M. Perrot propose par ailleurs une heuristique visant à ne considérer que des segments consécutifs dans une limite de déviation angulaire prédéfinie. Deux algorithmes sont alors proposés : l'un, appelé Poly-Isolines with Limited Deviation (PI-LD) et l'autre Poly-Isolines with Precomputed Directions (PI-PD). Ce dernier consiste à améliorer le premier en réduisant de 32^5 à 160 la quantité d'évaluations à effectuer. Le processus se divise alors en deux niveaux (deux noyaux), l'un réalisant la sélection du premier segment, l'autre évaluant les allongements successifs. Pour palier au problème de manque de robustesse de l'évaluation statistique dans certaines zones de pixels, appelées régions à faible pente (ce problème étant lié à la réduction du nombre d'évaluations), M. Perrot se propose d'appliquer une méthode hybride : pour ces régions un simple filtre « moyenneur » est alors utilisé. Les implémentations GPU des méthodes proposées sont ensuite évaluées. Il est un peu difficile de juger quel est la part de gain obtenue grâce aux heuristiques mises en œuvre et celle obtenue grâce à la parallélisation sur GPU. Le chapitre se termine en présentant une extension aux images de couleurs.

La troisième contribution porte sur les filtrages, d'une part « médian » et d'autre part « par convolution » implémentés sur GPU. Pour le filtre médian, la performance atteinte est essentiellement liée à l'optimisation du séquençement des instructions (pour garantir un usage du cache mémoire maximal) et à l'utilisation d'une méthode de calcul de médiane par « oubli » pour réduire la quantité de mémoire et ainsi permettre un usage maximal de registres dont l'accès est très rapide. Pour le filtre de convolution, l'optimisation réside essentiellement dans la réduction des accès mémoire, en effectuant des sommes partielles communes et partagées entre plusieurs pixels voisins. Il s'agit de contourner le problème classique des zones de recouvrement qui génèrent des accès mémoire redondants et concurrentiels. Notons que dans les deux cas, l'usage de « textures » permet de garantir une utilisation maximale du cache pour l'accès aux pixels de l'image à traiter.

Conclusion et avis général sur le travail de M. Perrot

Pour conclure, je pense que M. Perrot a effectué un très bon travail, tant en qualité qu'en quantité, pour une application importante du traitement d'images numériques : le traitement temps réel de gros volumes de données. Ses contributions sont à la fois fondamentales, par l'introduction de nouvelles heuristiques, que techniques, par le contrôle très fin des « threads » et des accès mémoire sur GPU. La gestion du passage à l'échelle et de la répartition des charges sont également des points très difficiles abordés par M. Perrot. Il a su y apporter des contributions certaines. Les résultats obtenus sont particulièrement encourageants et inciteront sans aucun doute à de nouvelles recherches, par exemple en introduisant l'usage de CPU multi-cœurs et l'usage de multiples GPU sur une seule carte mère, dans le but d'améliorer encore le parallélisme et le passage à l'échelle. La rédaction du manuscrit est claire et contient de nombreuses illustrations. Pour moi, il n'y a aucune réserve à ce que M. Perrot puisse présenter son travail en vue de l'obtention du doctorat de l'Université de Franche-Comté. Je suis très favorable à sa soutenance.

A handwritten signature in blue ink, which appears to be 'Jean-Michel Dischler', is written over the typed name and date.

Strasbourg, le 20 mars 2014,

Jean-Michel Dischler, Professeur d'informatique
ICube (UMR 7357), Université de Strasbourg