Comparaison de la consommation énergétique et du temps d'exécution d'un algorithme de traitement d'images optimisé sur des architectures SIMD et GPU

Andrea Petreto, Arthur Hennequin, Thomas Koehler, Thomas Romera, Yohan Fargeix, Boris Gaillard, Manuel Bouyer, Quentin Meunier, Lionel Lacassagne

Comparaison de la consommation énergétique et du temps d'exécution d'un algorithme de traitement d'images optimisé sur des architectures SIMD et GPU

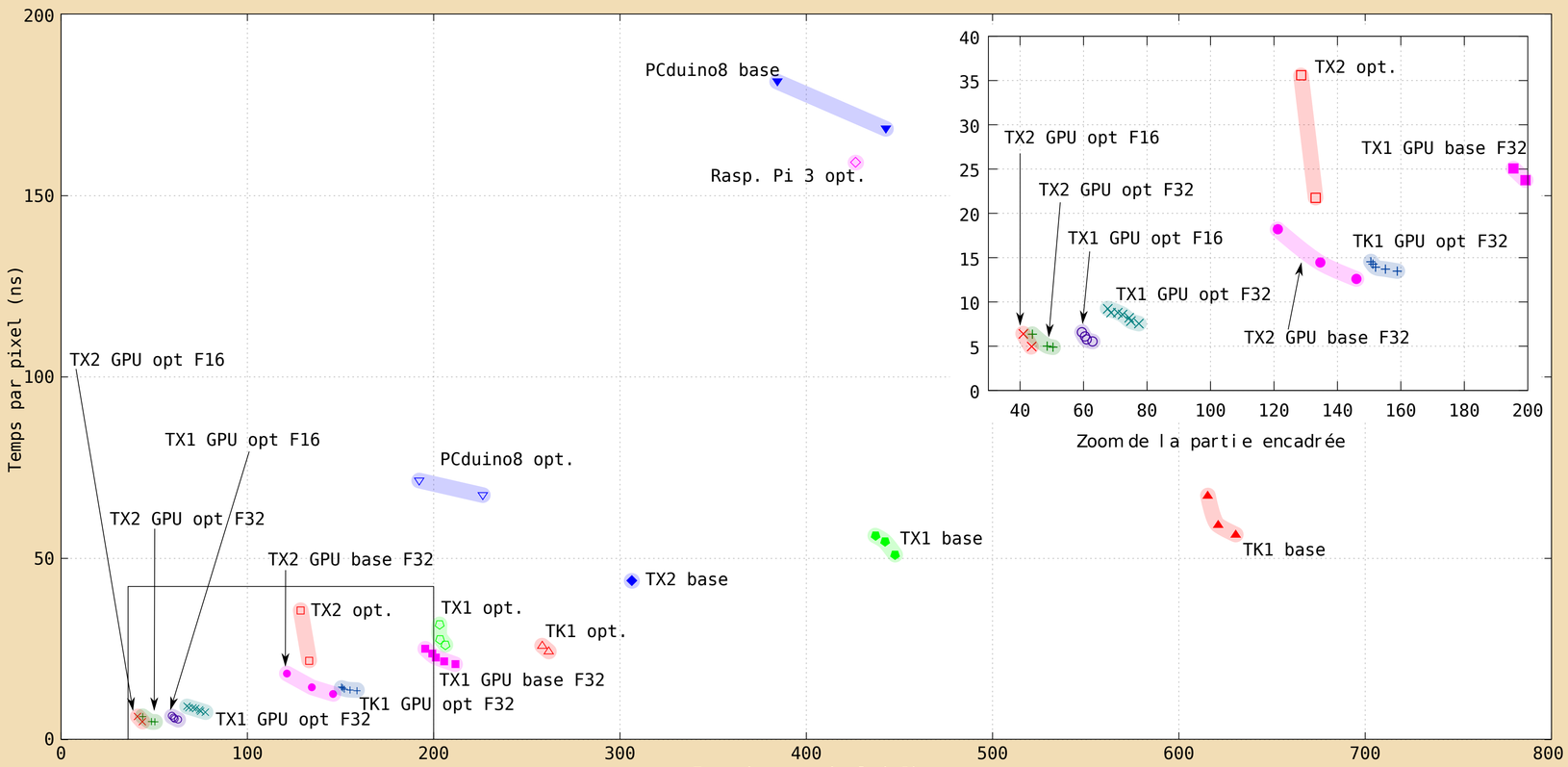
A. Petreto[1,2], A. Hennequin[1], T. Koehler[1], T. Romera[1], Y. Fargeix[1], B. Gaillard[2], M. Bouyer[1], Q. Meunier[1], L. Lacassagne[1]
Sorbonne Université, CNRS, LIP6 – Laboratoire d'Informatique de Paris 6 Paris, France [1]
{andrea.petreto,arthur.hennequin,thomas.koehler,thomas.romera}@lip6.fr
Lhéritier - Alcen – Cergy-Pontoise, France [2]
bgaillard@lheritier-alcen.fr

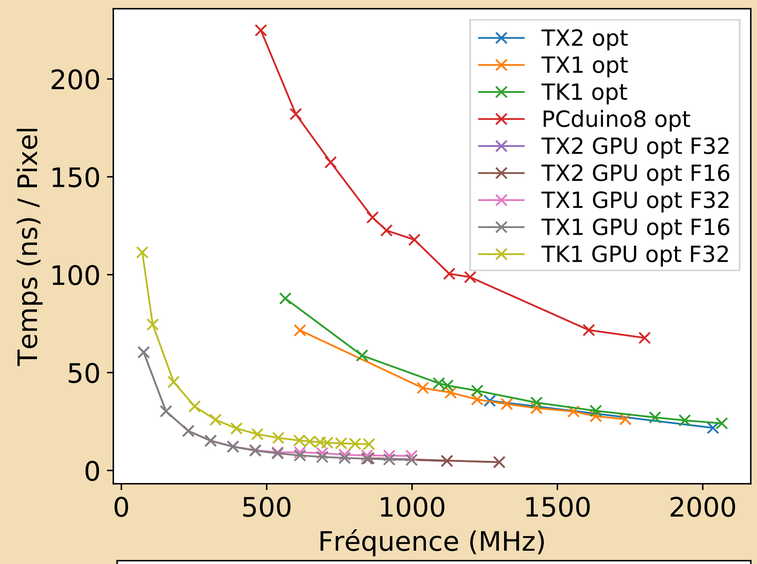
Résumé

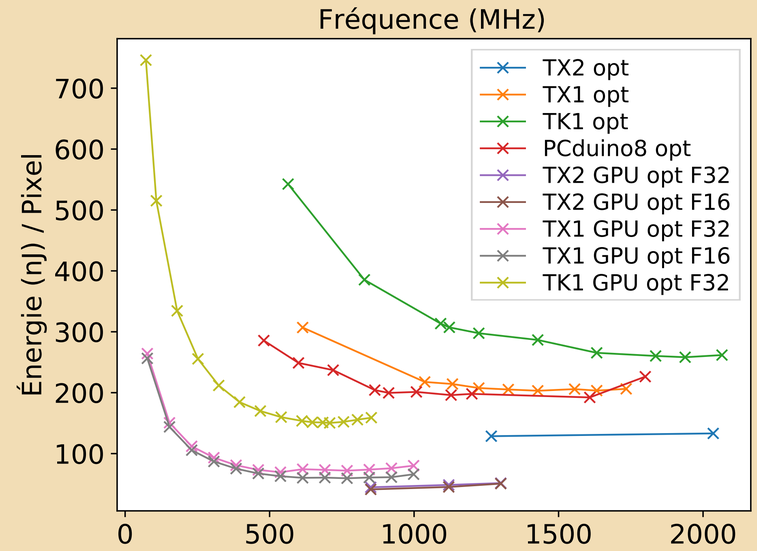
Ce poster présente et compare les implémentations optimisées d'un algorithme de **flot optique**, Horn-Schunck, sur des cartes embarquées à base de processeurs **SIMD** multicoeurs et de **GPU**. La comparaison est effectuée à la fois en termes de vitesse de calcul – pour atteindre une cadence de traitement **temps réel** – et en termes d'énergie. Les résultats obtenus montrent que les GPU sont les plus efficaces à la fois en termes de **vitesse** et de **consommation**, pouvant traiter dans la meilleure configuration 25 images de 8M pixels par seconde pour 0.35 joule par image.

Synthèse des résultats

carte	techno	CPU	Fmax (GHz)	GPU	Fmax (MHz)
PCduino8	28 nm	8×A7	1.80	-	-
Rasp. Pi 3	40 nm	4×A53	1.20	-	-
Jetson TK1	28 nm	4×A15	2.32	192 C Kepler	852
Jetson TX1	20 nm	4×A57	1.73	256 C Maxwell	998
Jetson TX2	16 nm	4×A57 (+ 2×Denver2)	2.00	256 C Pascal	1300

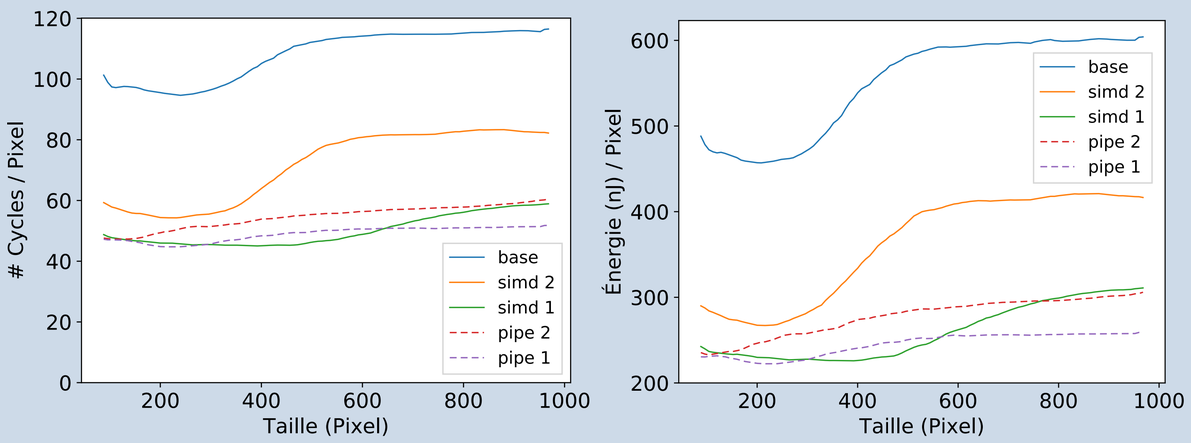






Frontière efficiente des fréquences de fonctionnement pour chaque architecture et algorithme étudié. La version *opt.* pour les CPU est la version la plus rapide (*pipe 1*). La configuration Rasp. Pi 3 *base* se trouve en dehors de l'espace représenté (énergie = 1145 nJ, temps = 415 ns).

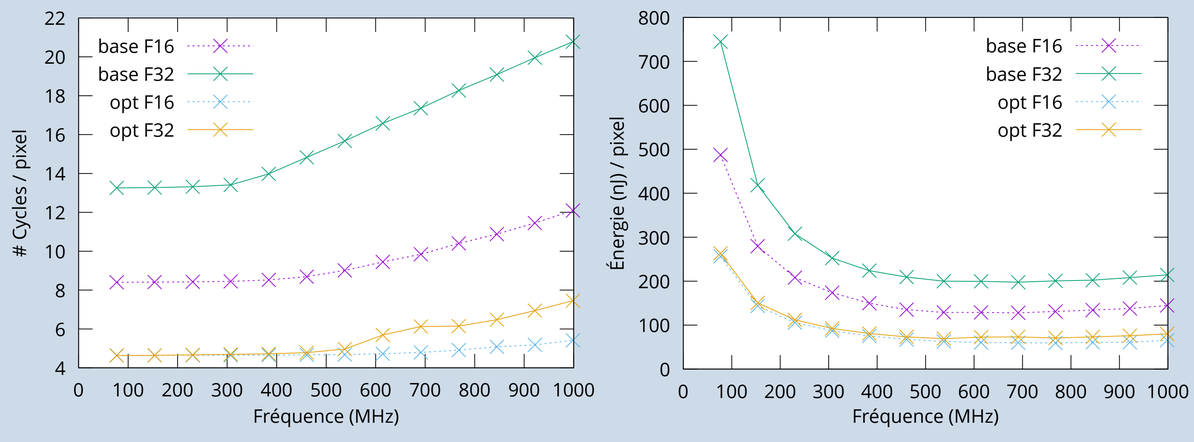
Vitesse et consommation sur CPU (TK1)



Version	OpenMP	SIMD	Pipeline	Mono-buffer
base	✓			
simd 1	✓	✓		✓
simd 2	✓	✓		
pipe 1	✓	✓	✓	✓
pipe 2	✓	✓	✓	

La version *simd 1* permet de reporter la sortie de cache pour des images plus grande. Les versions pipelinées *pipe 1* et *pipe 2* permettent de réduire le nombre d'accès hors du cache et réduisent donc la puissance consommée.

Vitesse et consommation sur GPU (TX1)



La version opt. réduit principalement les transferts mémoire : dans les basses fréquences elle est *Compute Bound* et F_{16} n'apporte pas de gain par manque d'*Instruction Level Parallelism* dans l'implémentation. Dans les hautes fréquences la cadence est réduite par manque de bande passante: elle est *Memory Bound* et F_{16} apporte un gain.

Conclusion

Ce poster présente une comparaison de plusieurs implémentations de l'algorithme de **Horn-Schunck**, servant à la détection de mouvement dans une image, sur différentes architectures **SIMD** et **GPU**, dans le but de réduire à la fois la **consommation** et le **temps de traitement**. Les configurations les plus efficaces permettent de traiter – à la cadence de 25 images/s – des images carrées de taille **2839 pixels sur GPU et 1355 sur CPU**. Parmi les travaux futurs, nous envisageons de regarder la précision des calculs au format virgule fixe 16 bits, afin de pouvoir doubler le parallélisme SIMD sur CPU. Enfin, nous visons une comparaison avec l'algorithme TV-L1.

Remerciements

Ce travail a été en partie subventionné par une thèse DGA, l'ESEP et Janus CNES. L'équipe Meteorix tient à remercier Tomoko Arai du projet PERC de l'Université de Chiba pour la fourniture de séquences vidéo, ainsi que Jean-Michel Morel et son équipe du CMLA de l'ENS Cachan.