Irregular & Architecture

Conception d'algorithmes parallèles efficaces pour des applications irrégulières sur des systèmes de vision embarqués hétérogènes

Encadrement Universitaire

— Lionel Lacassagne : lionel.lacassagne@lip6.fr, Pr. équipe Alsoc
— Julien Sopena : julien.sopena@lip6.fr, MCF équipe Delys

Résumé : Le but de cette thèse est de concevoir, implémenter et optimiser des algorithmes irréguliers de type *split & merge* manipulant des structures de données de type *union-find* pour des applications de vision embarquées sur des plateformes de type CPU+GPU. L'objectif est de concevoir des algorithmes de perception dite "pré-attentive" fournissant rapidement une pré-segmentation de la scène. Cette segmentation pourra être utilisée telle que, pour des systèmes low power ($\sim$1 watt) ou pourra servir d'entrée pour simplifier la complexité d'algorithmes de classification plus robustes et bien plus complexes, afin d'obtenir des systèmes embarqués de vision temps réel ayant une consommation limitée à une dizaine de watts.

Descriptif de la thèse

Contexte architectural : Aujourd'hui, les processeurs embarqués sont très majoritairement des processeurs multi-cœurs SIMD et il existe depuis quelques années des multi-GPU embarqués (cartes Nvidia Jetson, de 128 à 512 coeurs). Si cette nouvelle puissance de calcul ouvre de nombreuses perspectives pour les systèmes embarqués, elle pose aussi le défi du portage d'algorithmes s'exécutant traditionnellement sur de gros CPU ou de gros GPU. En effet, en comparaison, les ressources de calculs restent tout de même limitées. De plus, les contraintes de l'embarqué imposent une maîtrise de la consommation, problème particulièrement ardu sur ces architectures hybrides (CPU+GPU).

Contexte applicatif : Une des applications très prometteuses de ces nouvelles architectures est celle du traitement automatisé d'images dans des systèmes embarqués. Portée par les progrès en IA, on retrouve cette problématique dans de nombreux domaines industriels, et notamment celui des voitures autonomes. Malheureusement, les implémentations actuelles sont peu optimisées et pas assez rapides pour respecter la cadence d'acquisition de la caméra (typiquement 30 à 60 images par seconde) dite cadence temps réel.

Une observation rapide des méthodes en traitement d'images et vision par ordinateur montre que les algorithmes classiques de segmentation robuste comme *graph-cut* et *watershed* [6] sont peu à peu remplacés par des algorithmes encore plus robustes, mais bien plus complexes[10] [22]. Ainsi l'algorithme *Panoptic* apparu en 2019 [20] s'exécute – après optimisation – en 99 ms pour des images 1024×2048 sur un GPU Nvidia V100 consommant entre 250 et 300 watts [17]. Il a encore fallu simplifier le réseau et diviser la taille des images par 4 (512×1024) pour atteindre une cadence temps-réel de 30 FPS (*Frames Per Second*). L'implication de Toyota montre bien l'importance de la recherche de compromis qualité/consommation. Néanmoins, il manque encore deux ordres de grandeur pour tenir la contrainte de 60 FPS et 10 watts.

Afin de limiter la puissance de calcul nécessaire, et donc la consommation, il apparait *nécessaire* de concevoir des algorithmes bien plus légers pour des classes de systèmes plus petits. Ces algorithmes peuvent être utilisés seuls, lorsque les contraintes l'exigent ou être judicieusement positionné en amont d'une chaine de traitement plus robuste pour fournir une pré-segmentation et ne lancer ces derniers que sur des sous-ensembles de l'image.

Contexte algorithmique : Le but de cette thèse est de repenser une classe d'algorithmes manipulant des graphes enfouis dans les images, comme les algorithmes de *split & merge* et *union-find* réalisant une segmentation en région et un étiquetage en composantes connexes de chaque image.

Ces algorithmes très anciens [24, 26, 27, 16] sont irréguliers, car leur complexité et leur temps d'exécution dépendent de la nature de l'image. Cela est d'autant plus important que les images naturelles captées par une voiture présentent une très forte variabilité en termes de densité ou de granularité, avec une répartition très irrégulière. Ainsi, des algorithmes qui offrent de bonnes performances moyennes sur des images aléatoires peuvent devenir inefficaces sur des images naturelles.

Un problème aussi simple que la fusion itérative de listes dont les éléments dépendent de la nature d'une image peut devenir extrêmement complexe et aboutir à une fragmentation catastrophique de la mémoire (lié au problème de réallocation mémoire lorsqu'on dépasse la capacité allouée). Ainsi, si les algorithmes de *split* ont maintenant de bonnes propriétés [1, 23] (temps proportionnel au nombre de noeuds de l'arbre, et insensible à la surface des régions segmentées), ce n'est pas encore le cas des algorithmes de *merge*.

La recherche continue d'être active dans ce domaine et les algorithmes de *split & merge*, bien que moins précis que les algorithmes à base de *graph-cut* ou de *level sets* continuent d'être utilisés dans l'imagerie robotique et l'imagerie médicale [18, 21, 9, 25, 8, 13, 19, 28].

Résoudre un tel problème nécessite une double compétence en système et en architecture. Dans un premier temps il va s'agir de repenser les algorithmes et notamment les structures de données tout en tenant compte des caractéristiques très particulières des mémoires visées (notamment en termes d'hétérogénéité des temps d'accès). Puis dans un deuxième temps, il s'agira d'optimiser le placement et l'équilibrage de charge des différents threads, tout en minimisant la contention due à la synchronisation. Là encore, une double compétence sera nécessaire, pour à la fois optimiser le découpage de l'algorithme, mais aussi mettre en place des techniques d'ordonnancement spécifiques et des mécanismes de synchronisation efficaces comme l'utilisation de structure *wait-free* ou de verrouillage de type *RCU* (*Read-Copy-Update*).

Complémentarité et adéquation des compétences

Ce projet est à la frontière de l'architecture, du système et de la parallélisation. Il met en jeux plusieurs compétences des deux équipes impliquées. Parmi elles on retiendra :

dans l'équipe Alsoc (Lionel Lacassagne) : architecture [14], traitement d'images et géométrie discrète [1], structure *union-find* parallèle [4, 15] ;

dans l'équipe Delys (Julien Sopena) : gestion de la mémoire dans les systèmes embarqués avec contrainte de temps [2, 7], ordonnancement de tâches pour multicœurs [5, 3], la défragmentation mémoire sur des architectures NUMA [11, 12].

Références

[1] K. Aneja, F. Laguzet, L. Lacassagne, and A. Merigot. Video rate image segmentation by means of region splitting and merging. In *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 437–442, 2009.

[2] A. Blin, C. Courtaud, J. Sopena, J. L. Lawall, and G. Muller. Maximizing parallelism without exploding deadlines in a mixed criticality embedded system. In *28th Euromicro Conference on Real-Time Systems, ECRTS 2016, Toulouse, France, July 5-8, 2016*, pages 109–119, 2016.

[3] J. Bouron, S. Chevalley, B. Lepers, W. Zwaenepoel, R. Gouicem, J. Lawall, G. Muller, and J. Sopena. The battle of the schedulers : Freebsd ULE vs. linux CFS. In *2018 USENIX Annual Technical Conference, USENIX ATC 2018, Boston, MA, USA, July 11-13, 2018*, pages 85–96, 2018.

[4] L. Cabaret, L. Lacassagne, and D. Etiemble. Parallel Light Speed Labeling for connected component analysis on multi-core processors. *Journal of Real-Time Image Processing (JRTIP)*, 15(1) :173–196, 2018.

[5] D. Carver, R. Gouicem, J. Lozi, J. Sopena, B. Lepers, W. Zwaenepoel, N. Palix, J. Lawall, and G. Muller. Fork/wait and multicore frequency scaling : a generational clash. In *Proceedings of the 10th Workshop on Programming Languages and Operating Systems, SOSP 2019, Huntsville, ON, Canada, October 27-30, 2019*, pages 53–59, 2019.

[6] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watershed : A unifying graph-based optimization framework. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33,7 :1384–1399, 2010.

[7] C. Courtaud, J. Sopena, and G. Muller. Improving prediction accuracy of memory interferences for multicore platforms. In *28th IEEE Real-Time Systems Symposium, RTSS 2019, Hongkong, December 3-6, 2019*, 2019.

[8] R. Fa and A. K. Nandi. smart : novel self splitting-merging clustering algorithm. In *European Signal Processing Conference (EUSIPCO)*, pages 2198–2202, 2012.

[9] R. Fa and A. K. Nandi. An enhanced splitting-while-merging algorithm with finite mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3332–3336, 2013.

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35,8 :1915–1929, 2013.

[11] L. Gidra, G. Thomas, J. Sopena, and M. Shapiro. A study of the scalability of stop-the-world garbage collectors on multicores. In *Architectural Support for Programming Languages and Operating Systems, ASPLOS '13, Houston, TX, USA - March 16 - 20, 2013*, pages 229–240, 2013.

[12] L. Gidra, G. Thomas, J. Sopena, M. Shapiro, and N. Nguyen. Numagic : a garbage collector for big data on big NUMA machines. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '15, Istanbul, Turkey, March 14-18, 2015*, pages 661–673, 2015.

[13] Y. Gui, Z. Zheng, B. Xu, and Z. Zhang. An enhanced splitting-while-merging algorithm with finite mixture models. In *IEEE International Conference on Grey Systems and Intelligent Services*, pages 469–472, 2011.

[14] A. Hennequin, I. Masliah, and L. Lacassagne. Designing efficient SIMD algorithms for direct connected component labeling. In *ACM Workshop on Programming Models for SIMD/Vector Processing (PPoPP)*, pages 1–8, 2019.

[15] A. Hennequin, Q. L. Meunier, L. Lacassagne, and L. Cabaret. A new direct connected component labeling and analysis algorithm for GPUs. In *IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 1–6, 2018.

[16] S. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the ACM*, 23 :368–388, 1976.

[17] R. Hou, J. Li, A. Bhargava, A. Raventos, V. Guizilini, C. Fang, J. Lynch, and A. Gaidon. Real-time panoptic segmentation from dense detections. In *arXiv*, pages 1–12, 2019.

[18] N. R. Kasu and C. Saravanan. Segmentation on chest radiographs using otsu's and k-means clustering methods. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 210–213, 2018.

[19] D. Kelkar and S. Gupta. Improved quadtree method for split merge image segmentation. In *International Conference on Emerging Trends in Engineering and Technology*, pages 44–47, 2008.

[20] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar. Panoptic segmentation. In *Computer Vision and Pattern Recognitiopn (CVPR)*, pages 9404–9413, 2019.

[21] A. Kumar, P. Agham, R. Shanker, and M. Bhattacharya. Study of image segmentation techniques on microscopic cell images of section of rat brain for identification of cell body and dendrite. In *Springer, Information Systems Design and Intelligent Applications*, pages 452–462, 2018.

[22] P. Luc, C. Couprie, Y. Lecun, and J. Verbeek. Panoptic segmentation. In *Computer Vision and Pattern Recognitiopn (CVPR)*, pages 9404–9413, 2019.

[23] A. Merigot. Revisiting image splitting. In *International Conference on Image Analysis and Processing (ICIAP)*, page 0, 2003.

[24] A. Rosenfeld and J. Platz. Sequential operator in digital pictures processing. *Journal of ACM*, 13,4 :471–494, 1966.

[25] P. Roy, D. Das, and P. K. Biswas. Real time vlsi implementation of a fast split and merge segmentation algorithm. In *IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–8, 2012.

[26] R. Tarjan. Efficiency of good but not linear set union algorithm. *Journal of ACM*, 22,2 :215–225, 1975.

[27] R. Tarjan and J. Leeuwen. Worst-case analysis of set union algorithms. *Journal of ACM*, 31 :245–281, 1984.

[28] G. Xuejing, C. Linlin, and Y. Kangze. Two parallel strategies of split-merge algorithm for image segmentation. In *International Conference on Wavelet Analysis and Pattern Recognition*, pages 840–845, 2007.