innovatives

VOITURE DU FUTUR

Usages

Image Processing

Embedded Systems

Prédiction des performances d'algorithmes ADAS sur architectures embarquées

Romain SAUSSARD[1,2] – Boubker BOUZID[1] – Roger REYNAUD[2] – Marius VASILIU[2]

[1]Renault, [2]Université Paris Sud

Introduction

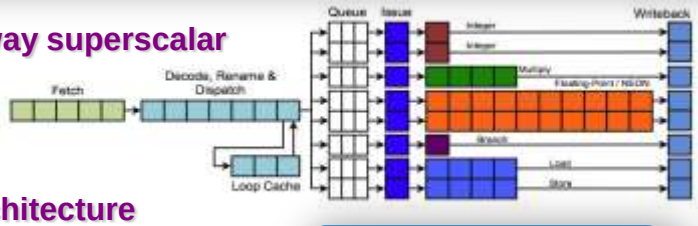
- Computer Vision Algorithms are widely used in automotive field for the ADAS and smart car applications.
- A lot of computing architectures can be used to embed those algorithms : ARM, DSP, GPU and heterogeneous one like the K1. It's not easy to choose the best algorithm – architecture association.
- Existing models for performance prediction are only applicable to classical architecture, not on heterogeneous systems (for example, algorithm proposed by [1] can be used only for CUDA).
- We propose a method to predict performance on multiple, heterogeneous architectures in order to help choosing the best algorithm – architecture association.
- We illustrate our approach with a lane detection algorithm predicted then embedded on different architectures.

Classes of Instructions

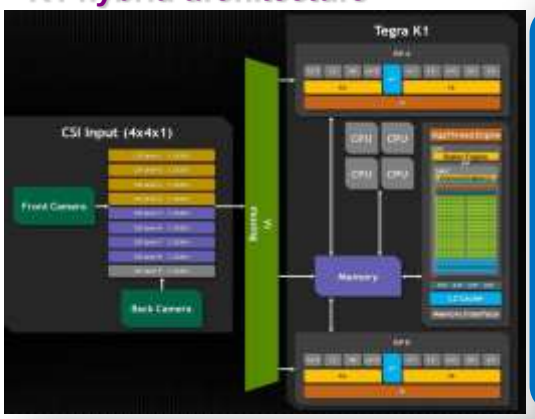
- An algorithm is a set of instructions, and each instruction can be classified.
- An architecture, a, has different throughput ($p_{c,a}$ in instructions per cycle) for each instruction class, c.
- C : set of computing instructions, $c \in C$.
- M : memory instructions (*Load & Store*).
- Ia : arithmetic intensity [2], number of operations for each memory instruction. This can be used to

$$Ia = \frac{N_C}{N_M} \qquad t_{max,a} = \sum_{i \in C} \frac{N_c}{p_{c,a}} \qquad t_{min,a} = \max_{\{i \in C\}} \left(\frac{N_c}{p_{c,a}} \right)$$

ARM A15 3-way superscalar architecture



K1 hybrid architecture



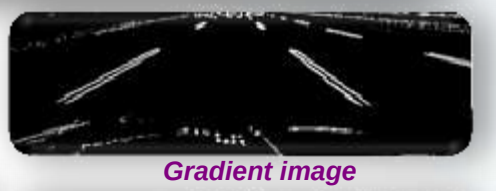
Throughput : $p_{c,a}$

Instruction	ARM A15	CUDA/GPU K1
Simple Int	2	160
Mult. Int	1	32
Float	1	192
Specific	*	32
Branch	1	32
Address	1	160

•Multiple instructions

Lane detection algorithm

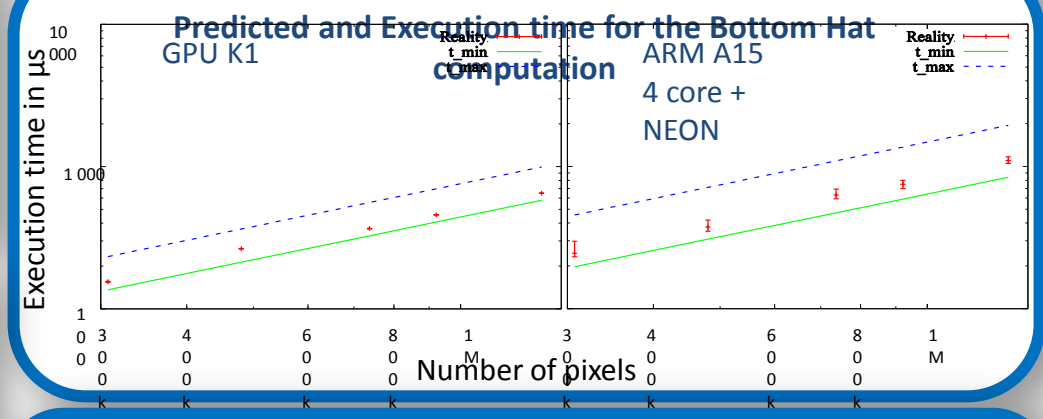
Input image



Gradient image

Bottom Hat (1x5 SE)

Predicted and Execution time for the Bottom Hat computation



Predicted Results

Algorithm		ARM A15 1.5 GHz 1 core	ARM A15 1.5 GHz 4 cores + NEON	CUDA/ GPU K1 600 MHz
Gradient $Ia = 4$	t_{max}	19 Mp/s	380 Mp/s	490 Mp/s
	t_{min}	57 Mp/s	1030 Mp/s	900 Mp/s
	Reality	32 Mp/s	450 Mp/s	660 Mp/s
Bottom Hat $Ia = 5$	t_{max}	18 Mp/s	670 Mp/s	1320 Mp/s
	t_{min}	32 Mp/s	1560 Mp/s	2260 Mp/s
	Reality	30 Mp/s	1250 Mp/s	2000 Mp/s

Applications and Future Work

- Our model is able to predict an execution time interval for heterogeneous architectures if the Ia is high enough.
- This approach can help automotive industry to embed complex image processing applications (e.g. autonomous driving)
- Future steps : generalize & test our model for any complex algorithms, on any embedded / hybrid architecture.

[1] HONG, Sunpyo et KIM, Hyesoon. An analytical model for a GPU architecture with memory-level and thread-level parallelism awareness. In : *ACM SIGARCH Computer Architecture News*. ACM, 2009. p. 152-163.

[2] M. Harris. Mapping computational concepts to GPUs. In *ACM SIGGRAPH 2005 Courses*, page 50. ACM, 2005.

[3] S. Williams, A. Waterman, and D.Patterson. Roofline : an insightful visual performance model for multicore architectures. *Communications of the ACM* (CACM), 52(4):65–76, 2009.

Université Paris Sud / IEF et Renault S.A.

www.cnrs.fr

UNIVERSITÉ PARIS SUD

RENAULT

IFSTTAR