Universidade de Aveiro

Mestrado Integrado em Engenharia Computacional

Mestrado em Engenharia Computacional

Computação Paralela

Project 1:
FAST Corner Detection with CUDA and OpenMP

Academic year: 2022/2023 Nuno Lau/Rui Costa

1. Introduction

In this assignment gray scale images will be processed in order to detect the position of corners. An image will be modelled as an array of integers which values range from 0 to 255. The values in the image specify the pixel luminance, hence a value of 0 indicates a black pixel and a value of 255 indicates a white pixel. An image will be stored in memory as an array (or matrix) of integer values where each element of the array/matrix corresponds to a pixel in the image.

The image will be processed for corner detection using the FAST Detector [1]. This algorithm considers as corners pixels that have a certain number of consecutive neighbors (along a circle of radius 3) that are either darker or brighter than the corner by a certain amount. The algorithm may be enhanced by associating a score to each detected corner and keeping only corners which score are local maximum. The FAST Detector has three parameters: the number of consecutive pixels to count and the minimum difference to be darker (or brighter) and wether non local maximum are eliminated or not.

2. Work description

The objective of this work is to start from the source code package **`cp_fastDetector.tgz`** (available at moodle) and develop improved versions of the FAST detector using the CUDA and OpenMP platforms. Images may be of any size. The function **`fastDetectorDevice()`** should encapsulate all the operations of preparation, execution and result retrieval of the CUDA kernel. The function **`fastDetectorOpenMP()`** should include the OpenMP implementation of the detector. The assignment can be tested using the **`banana.ua.pt`** computer that includes a GPU with compute capability 7.5, but you are advised to also use other computers for testing. The function **`fastDetectorHost()`** and its subfunctions should not be changed.

You may develop (and compare) several versions of your code that use different functionalities of the CUDA device (global memory, shared memory, texture memory, etc.). If you do test the use of different CUDA memory resources, please deliver all developed versions and use an archive file with an additional suffix in its name (ex: **proj1_*nm1*_*nm2*_shared.tgz**[1]) for the different memory types that were used.

3. Important notes

Each group must deliver:

[1] "**`nm1`**" and "**`nm2`**" are to be replaced by your UA id numbers.

- the source code of the developed programs;
- a report that presents: a) the general architecture of the developed solutions; b) the main data structures and algorithms that have been used; c) the results that have been attained; d) basic instructions for compilation and execution of your program.

During the development of this assignment you should follow an ethical conduct that prohibits plagiarism, in any form, as well as the participation of external elements in the assignment development. Any initiative that, judged by the teaching team, might be considered as a plagiarism situation will have real consequences on the student(s) evaluation and may lead to disciplinary sanctions.

4. Due dates

- **May 22, 2023**

Submitting your work after the due date will be penalized with 1 point less for each day of delay.

Bibliography:

[1] Machine Learning for High-Speed Corner Detection. Rosten, E., Drummond, T. (2006). In: Computer Vision – ECCV 2006. ECCV 2006. LNCS, vol 3951. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11744023_34

[2] *NVIDIA CUDA C++ Programming Guide*, v12.1, NVIDIA (available at elearning)

[3] *CUDA C++ BEST PRACTICES GUIDE*, v12.1, NVIDIA (available at elearning)