

Lập trình song song trên GPU

BT1: CUDA cơ bản



Nên nhớ mục tiêu chính ở đây là **học, học một cách chân thật**. Bạn có thể thảo luận ý tưởng với bạn khác, nhưng **bài làm phải là của bạn, dựa trên sự hiểu thật sự của bạn**. **Nếu vi phạm thì sẽ bị 0 điểm cho toàn bộ môn học**.

Đề bài

Viết chương trình chuyển ảnh RGB sang ảnh Grayscale (mình đã hướng dẫn sơ bộ ở trên lớp thực hành).
Bạn sử dụng công thức chuyển:

giá-trị-grayscale = $0.299 \times \text{giá-trị-red} + 0.587 \times \text{giá-trị-green} + 0.114 \times \text{giá-trị-blue}$

Mình có đính kèm file ảnh đầu vào “in.pnm” và file ảnh đầu ra đúng “correct_out.pnm” (trong Windows, bạn có thể xem file *.pnm bằng chương trình [IrfanView](#)). Mình cũng đã viết sẵn khung chương trình trong file “bt1.cu” đính kèm. Chương trình sẽ đọc file ảnh đầu vào RGB, chuyển ảnh đầu vào RGB sang ảnh đầu ra Grayscale (tuần tự bằng host hoặc song song bằng device), ghi ảnh đầu ra xuống file, và so sánh ảnh đầu ra này với ảnh đầu ra đúng. Bạn sẽ cần phải viết code ở những chỗ mình để “// TODO” [bạn sẽ phải viết code để: (i) thực hiện chuyển đổi ảnh một cách tuần tự bằng host, (ii) truy vấn thông tin device, (iii) thực hiện chuyển đổi ảnh một cách song song bằng device]. **Lưu ý: file “bt1.cu” này có thay đổi so với file trên lớp thực hành (ví dụ, mình có tích hợp luôn phần kiểm tra kết quả vào file “bt1.cu”); bạn lấy lại file “bt1.cu” này để làm bài.**

Hướng dẫn về các câu lệnh:

- Biên dịch file “bt1.cu”: `nvcc bt1.cu -o bt1`
- Chạy file “bt1.exe” (trong Linux thì là file “bt1”) với file ảnh đầu vào là “in.pnm”, file ảnh đầu ra là “out1.pnm”, file ảnh đầu ra đúng là “correct_out.pnm”, và dùng CPU (host) để tính toán:
`.\bt1.exe in.pnm out1.pnm correct_out.pnm cpu`
(Linux: `./bt1 in.pnm out1.pnm correct_out.pnm cpu`)
Kết quả so sánh giữa ảnh đầu ra của bạn và ảnh đầu ra đúng là giá trị khác biệt trung bình giữa 2 mảng pixel của 2 ảnh (được tính bằng cách: lấy các pixel tương ứng của 2 ảnh trừ cho nhau, lấy trị tuyệt đối, và cuối cùng tính trung bình hết lại). Như vậy, nếu giá trị khác biệt trung bình bằng 0 thì nghĩa là 2 ảnh giống hệt nhau. [Nếu giá trị khác biệt trung bình có giá trị nhỏ (ví dụ, 0.xxx) thì chưa chắc là sai, vì khi tính toán số thực thì giữa CPU và GPU có thể có sai biệt nhỏ; nhưng nếu giá trị khác biệt trung bình lớn hơn 0.xxx thì chắc là sai rồi.]
- Chạy file “bt1.exe” (trong Linux thì là file “bt1”) với file ảnh đầu vào là “in.pnm”, file ảnh đầu ra là “out2.pnm”, file ảnh đầu ra đúng là “correct_out.pnm”, và dùng GPU (device) để tính toán:
`.\bt1.exe in.pnm out2.pnm correct_out.pnm gpu`
(Linux: `./bt1 in.pnm out2.pnm correct_out.pnm gpu`)

Mặc định thì sẽ dùng block có kích thước 32×32; nếu bạn muốn chỉ định kích thước block thì truyền thêm vào câu lệnh 2 con số lần lượt ứng với kích thước theo chiều x và theo chiều y của block (ví dụ, `./bt1.exe in.pnm out2.pnm correct_out.pnm gpu 32 16`).

Bạn sẽ ghi nhận lại các kết quả chạy trong file báo cáo (ở phần header của file bạn ghi họ tên và MSSV):

- Chụp lại hình kết quả chạy khi sử dụng CPU và khi sử dụng GPU. Ví dụ:

```
teacher@2adff12762339:~$ ./bt1 in.pnm out1.pnm correct_out.pnm cpu
Image size (width x height): 512 x 512
Processing time: 3.566592 ms
The average pixel difference between 'out1.pnm' and 'correct_out.pnm': 0.000000
teacher@2adff12762339:~$ ./bt1 in.pnm out2.pnm correct_out.pnm gpu
Image size (width x height): 512 x 512
GPU name: TITAN Xp
GPU compute capability: 6.1
Block size: 32 x 32
Grid size: 16 x 16
Processing time: 2.189344 ms
The average pixel difference between 'out2.pnm' and 'correct_out.pnm': 0.000000
```

- Chụp lại hình kết quả chạy khi sử dụng GPU với các kích thước block khác nhau: 8x8, 16x16, 32x32, 64x64. Ví dụ:

```
teacher@2adff12762339:~$ ./bt1 in.pnm out2.pnm correct_out.pnm gpu 8 8
Image size (width x height): 512 x 512
GPU name: TITAN Xp
GPU compute capability: 6.1
Block size: 8 x 8
Grid size: 64 x 64
Processing time: 1.765568 ms
The average pixel difference between 'out2.pnm' and 'correct_out.pnm': 0.000000
teacher@2adff12762339:~$ ./bt1 in.pnm out2.pnm correct_out.pnm gpu 16 16
Image size (width x height): 512 x 512
GPU name: TITAN Xp
GPU compute capability: 6.1
Block size: 16 x 16
Grid size: 32 x 32
Processing time: 2.038208 ms
The average pixel difference between 'out2.pnm' and 'correct_out.pnm': 0.000000
teacher@2adff12762339:~$ ./bt1 in.pnm out2.pnm correct_out.pnm gpu 32 32
Image size (width x height): 512 x 512
GPU name: TITAN Xp
GPU compute capability: 6.1
Block size: 32 x 32
Grid size: 16 x 16
Processing time: 1.723744 ms
The average pixel difference between 'out2.pnm' and 'correct_out.pnm': 0.000000
teacher@2adff12762339:~$ ./bt1 in.pnm out2.pnm correct_out.pnm gpu 64 64
Image size (width x height): 512 x 512
GPU name: TITAN Xp
GPU compute capability: 6.1
Block size: 64 x 64
Grid size: 8 x 8
Error: bt1_sol.cu:232, code: 9, reason: invalid configuration argument
```

(Trên máy của mình, mình quan sát thấy là với cùng một kích thước block, thời gian chạy có thể thay đổi chút ít từ lần chạy này qua lần chạy khác. Nếu muốn so sánh chính xác hơn về thời gian chạy thì với mỗi kích thước block, cần chạy một số lần rồi tính trung bình. Tuy nhiên, ở bài này, không yêu cầu bạn phải làm như vậy; chủ yếu là để bạn thấy các kích thước block khác nhau sẽ đưa đến các thời gian chạy khác nhau, và để xem chương trình của bạn có bắt được lỗi khi dùng kích thước block không hợp lệ không).

Nộp bài

Trong thư mục <MSSV> (vd, nếu bạn có MSSV là 1234567 thì bạn đặt tên thư mục là 1234567), bạn để:

- File code “bt1.cu”

- File báo cáo “bt1.pdf”

Sau đó, bạn nén thư mục này lại và nộp ở link trên moodle.