



Durée : 2h — Documents autorisés

Programme OpenMP & CUDA — 14 points

1- Le constructeur d'appareils photos JiFu vient de mettre au point un nouveau type de capteur permettant d'améliorer la qualité des photos prises par ses appareils.

14pts

Ce capteur utilise des « filtres de Bayer » modifiés.

Précision

Le « filtre de Bayer » permet de recomposer une information de couleur grâce à la combinaison de valeurs de 4 photosites placés derrière des micro-lentilles de couleurs Rouge, Vert et Bleu. Un « photosite » est une surface sensible à la lumière, permettant d'obtenir l'intensité lumineuse de la lumière qui éclaire cette surface. Il ne fournit qu'une information en noir & blanc. Grâce à une micro-lentille de couleur, il peut alors fournir l'intensité lumineuse correspondant à la couleur de cette micro-lentille.

Le capteur est constitué de millions de photosites correspondant, chacun, à un pixel dans l'image construite par l'appareil photo.

Ce capteur est constitué d'un pavage de « filtre de Bayer », soit régulier (colonne de gauche), soit « modifié » (colonne de droite) :

Le « filtre de Bayer », avec un photosite sensible au Rouge, un autre au Bleu et deux au Vert :

Table with 2 columns and 2 rows: R V, V B

Le « pavage » sur tout le capteur :

Grid of Bayer filters: R V R V ...; V B V B ...; R V R V ...; V B V B ...; vertical dots

Le « filtre de Bayer modifié » consiste à répartir de manière aléatoire les couleurs R, V, V et B :

Table with 2 columns and 2 rows: x w, y z

Ce qui donne un pavage aléatoire sur le capteur :

Grid of modified Bayer filters: R V R B ...; B V V V ...; B V R B ...; V R V V ...; vertical dots

Le constructeur voudrait faire du contrôle qualité de chaque capteur qu'il fabrique : vérifier qu'il y ait bien les 4 filtres R, V, V et B dans chacun des « filtres de Bayer modifiés ». Exemple :

Grid of modified Bayer filters: R V R B ...; B V V V ...; B V R B ...; V R V V ...; vertical dots

=> capteur correct !

Grid of modified Bayer filters: R V R B ...; V V V V ...; B V R B ...; V R V V ...; vertical dots

=> capteur incorrect ! Il y a un filtre V de trop sur le « filtre de Bayer » en haut à gauche.

Pour réaliser ce travail de vérification, une matrice de la dimension du capteur est fournie sous forme d'un tableau de type char.

Par exemple pour un capteur de 16M pixels, un tableau `Capteur[4096][4096]` est fourni avec, dans chaque case, le caractère 'R', ou 'V' ou 'B'.

Questions :

- a. Un capteur est considéré correct, si pour tout « filtre de Bayer » qu'il contient, c-à-d dans tout bloc de 2x2, il y a bien 1 caractère 'R', 2 caractères 'V' et 1 caractère 'B'. (2pts)

Le contrôle de qualité exige que si un seul « filtre de Bayer » est mauvais, le capteur entier soit considéré comme mauvais.

Proposez une **structure de données** et/ou une **structure de contrôle** permettant :

- ◇ la parallélisation du processus de vérification ;
- ◇ l'obtention du résultat final : tous les « filtres de Bayer » correct = capteur correct.

Vous discuterez et justifierez votre proposition.

- b. Proposez une méthode de parallélisation avec comme cible d'exploitation : (2pts)

- ◇ une version CPU ;
- ◇ un nombre variable de cœurs : vous discuterez de la « scalability » de votre proposition.

- c. Écrire un programme OpenMP permettant de contrôler la qualité d'un capteur 16M pixels en exploitant au mieux les possibilités de traitement parallèle. La constitution de ce capteur est donnée dans le tableau char `Capteur[4096][4096]`. (4pts)

- d. On veut maintenant faire une version GPU de la vérification du capteur. Proposez une méthode permettant de faire le maximum du traitement à l'intérieur du GPU. (2pts)

Quel travail va réaliser chaque *thread* ?

Comment récupérer le résultat « Capteur correct » ou « Capteur incorrect » sur le « host » ?

Est-il possible de tirer parti de la mémoire partagée d'un « block » ?

Vous donnerez et justifierez la « grille » à utiliser pour faire cette parallélisation compte tenu des limitations du traitement GPU.

- e. Écrire un programme CUDA réalisant ce travail. La constitution de ce capteur est donnée dans le tableau char `Capteur[4096][4096]`. (4pts)

Rappel sur les accès tableau en C

```
1 int a[N][N];
2 int i = 2;
3 int j = 3;
4 a[i][j] = 5;
5 printf("%d\n", a[j+i*N]);
```

Les notations :
 `a[i][j]`
 `a[j+i*N]`
sont équivalentes.

■■■■ Question de cours — 6 points

2– Expliquez comment est traitée l'instruction « if (condition) then traitement_1 else traitement_2 » dans une architecture SIMD, comme celle du processeur CELL de la Playstation 3, et pourquoi il n'y a pas d'autre solution possible. **2pts**

3– Qu'est-ce que le « recouvrement calcul/communication » ?
2pts Pourquoi n'est-il pas toujours intéressant de découper un problème suivant le plus grand nombre de tâches ?

4– Est-ce que la programmation GPGPU proposée par CUDA est plus proche de la directive de parallélisation OpenMP : **2pts**

- `#pragma omp parallel for`
- `#pragma omp parallel sections`

Vous justifierez votre réponse.