Ειδικά Θέματα Παράλληλου Προγραμματισμού 2023-2024

Εργασία #3 (OpenCL)

Το αντικείμενο της εργασίας είναι η φόρτωση και επεξεργασία εικόνων. Κάθε εικόνα αποτελείται από έναν αριθμό pixels όπου το χρώμα του κάθε πίξελ αναπαριστάται από τις τιμές των 3 βασικών χρωμάτων *Red*, *Green*, *Blue*. Συνήθως σε αυτά τα χρωματικά κανάλια προσθέτουμε και ένα τέταρτο κανάλι, *Alpha*, το οποίο αναπαριστά το ποσοστό της διαφάνειας του pixel. Τα κανάλια αποθηκεύονται ώς μεταβλητές τύπου *unsigned char*, που παίρνουν τιμές από το *0* (καθόλου συνεισφορά) έως το *255* (μέγιστη συνεισφορά). Η κάθε εικόνα αποθηκεύεται ως ένας πίνακας από unsigned chars μεγέθους [width * height * number_of_channels].

Α. Θόλωση εικόνας

Στην πρώτη εργασία (*HW1*) σας είχε δοθεί η συνάρτηση *gaussian_blur_separate_serial()*, η οποία εφαρμόζει την τεχνική δύο περασμάτων από Gaussian Blur ώστε να επιταχύνει το αποτέλεσμα της θόλωσης της εικόνας “street_night.jpg”. Σας ζητείται να δημιουργήσετε μία νέα συνάρτηση (gaussian_blur_separate_parallel()), στην οποία να φορτώνεται η ίδια εικόνα και έπειτα να γίνεται η θόλωση παράλληλα, χρησιμοποιώντας την OpenCL ώς εξής:

- Δημιουργήστε ένα OpenCL context με ένα GPU device. Σε περίπτωση που δεν είναι διαθέσιμο, χρησιμοποιήστε ένα CPU device.
- Γράψτε τον κώδικα του blur σε OpenCL C Language και αποθηκεύστε τον σε ένα αρχείο με όνομα *kernel.cl*.
- Παρατηρήστε ότι τα blur weights είναι ίδια και ξανα-υπολογίζονται σε κάθε pixel που εφαρμόζεται το φίλτρο. Προϋπολογίστε τα blur weights σε host κώδικα, περάστε τα σε ένα OpenCL buffer και χρησιμοποιήστε τα κατευθείαν από τον device κώδικα (ώστε να μην χρειαστεί να γίνει η πράξη υπολογισμού των weights στον kernel).
- Εκτελέστε τον πυρήνα δύο φορές, μία για κάθε κάθετο και οριζόντιο πέρασμα, ώστε να υπολογιστεί η θολωμένη εικόνα.
- Εγγραφή της θολωμένης εικόνας σε ένα αρχείο με όνομα “image_blurred_final.jpg”
- Πειραματιστείτε με διάφορες τιμές για το local_work_size. Το μέγεθος της ομάδας εργασίας πρέπει να επιλεγεί έτσι ώστε να μεγιστοποιεί τη χρήση των πόρων στην κάρτα γραφικών και να ελαχιστοποιεί την καθυστέρηση πρόσβασης στη μνήμη.

Απαιτούμενα

- Ο πηγαίος κώδικας που δίνετε για τις υλοποιήσεις σας θα πρέπει να είναι σωστά δομημένος, στοιχισμένος και σχολιασμένος (προτεινόμενη γλώσσα τα Αγγλικά).
- Θα πρέπει να παραδώσετε πλήρη αναφορά, περιλαμβάνοντας και γραφικές παραστάσεις χρονομετρήσεων καθώς και συζήτηση γύρω από τα αποτελέσματα. Στην αναφορά θα πρέπει να εμφανίζεται το όνομα σας και ο αριθμός μητρώου.
- Θα πρέπει να παραδώσετε την εικόνα που δημιουργήθηκε από την εκτέλεση του προγράμματος.
- Τα προγράμματά σας (πηγαίοι κώδικες + αναφορά + εικόνα) θα πρέπει να τα παραδώσετε στο eclass του μαθήματος σε μορφή zip αρχείου. Στο όνομα του αρχείου θα πρέπει να περιλαμβάνεται ο αριθμός μητρώου του φοιτητή.
- Οι ασκήσεις ελέγχονται για κοινό κώδικα και αντιγραφή. Τέτοιες περιπτώσεις φυσικά θα μηδενίζονται και δεν θα υπάρχει δικαίωμα εξέτασης στην εξεταστική περιόδο.
- Για τη χρονομέτρηση μπορείτε να χρησιμοποιήσετε τις κλήσεις χρονομέτρησης στην C++ std::chrono::high_resolution_clock::now().
- Για κάθε περίπτωση, ένα πρόγραμμα θα εκτελείται τουλάχιστον 4 φορές και ο τελικός χρόνος θα είναι ο μέσος όρος των τεσσάρων χρόνων.

Παρατηρήσεις

- Η τελική βαθμολογία θα παρθεί μετά από προφορική εξέταση. Σχετικό πρόγραμμα εξέτασης θα βγει εγκαίρως μετά την παράδοση της εργασίας στην ιστοσελίδα του μαθήματος.
- Για οποιοδήποτε πρόβλημα εγκατάστασης της OpenCL στο μηχάνημα σας, παρακαλώ επικοινωνήστε με τον διδάσκοντα (agkar@aueb.gr).

Προθεσμία παράδοσης: Κυριακή, 2 Ιουνίου 2024

Καλή Επιτυχία!
Αναστάσιος Γκαραβέλης