Ειδικά Θέματα Παράλληλου Προγραμματισμού 2024-2025

Εργασία 1 (pthread/std::thread)

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 (40%)

Σας δίνεται η συνάρτηση *gaussian_blur_serial()*, η οποία εφαρμόζει σειριακά το φίλτρο Gaussian Blur προκειμένου να θολώσει (ή να ομαλοποιήσει) μία εικόνα "garden.jpg". Η συνάρτηση φορτώνει την εικόνα σε έναν πίνακα img_in, παράγει τη θολωμένη της εκδοχή img_out, βάσει μίας ακτίνας θόλωσης *KERNEL_RADIUS* (όσο μεγαλύτερη η ακτίνα, τόσο πιο έντονο το θόλωμα) και την αποθηκεύει σε ένα αρχείο τύπου JPG. Σας ζητείται να δημιουργήσετε μία νέα συνάρτηση (gaussian_blur_parallel()), στην οποία να φορτώνεται η ίδια εικόνα και έπειτα να γίνεται η θόλωση παράλληλα, χρησιμοποιώντας είτε τα pthreads είτε τα C++ threads. Θα πρέπει να πειραματιστείτε με το μοίρασμα της δουλειάς σε 2, 4 και 8 threads και να χρονομετρήσετε την απόδοση τους και να τα συγκρίνετε μεταξύ τους αλλά και με τη σειριακή έκδοση. Τέλος να αποθηκεύσεται την εικόνα σε ένα αρχείο με όνομα "blurred_image_parallel.jpg".

Β. Ενίσχυση και θόλωση εικόνας (60%)

Σας δίνεται 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" (**Προσοχή**: Το κάθε πέρασμα βασίζεται στο αποτέλεσμα του προηγούμενου περάσματος). Σας ζητείται να δημιουργήσετε, με βάση την σειριακή συνάρτηση, μια νέα συνάρτηση με όνομα gaussian_blur_separate_parallel(). Η συνάρτηση θα φορτώνει την ίδια εικόνα και θα δημιουργεί 4 threads, τα οποία θα είναι ενεργά έως το τέλος του προγράμματος και στα οποία θα πρέπει να εκτελεστούν παράλληλα οι εξής εργασίες:

1. Υπολογισμός της μέγιστης τιμής των 4 channels ξεχωριστά, από όλα τα pixel της εικόνας.
2. Κανονικοποίηση των τιμών του κάθε channel σε κάθε pixel της εικόνας με βάση την μέγιστη τιμή του channel. Η κανονικοποίηση θα γίνεται στο εύρος τιμών που παρέχει ο τύπος unsigned char (0-255). Η κανονικοποίηση μπορεί να γίνει με βάση τον τύπο:

pixel[channel] = 255 * pixel[channel] / maxValue[channel];

3. Εγγραφή της κανονικοποιημένης εικόνας σε ένα αρχείο με όνομα “image_normalized.jpg” [Μόνο από ένα thread]
4. Θόλωση της εικόνας στον οριζόντιο άξονα
5. Εγγραφή της θολωμένης εικόνας σε ένα αρχείο με όνομα “image_blurred_horizontal.jpg”. [Μόνο από ένα thread]
6. Θόλωση της εικόνας στον κάθετο άξονα
7. Εγγραφή της θολωμένης εικόνας σε ένα αρχείο με όνομα “image_blurred_final.jpg”. [Μόνο από ένα thread]

Χρησιμοποιήστε τις κατάλληλες μεθόδους αμοιβαίου αποκλεισμού και συγχρονισμού ώστε να επιτευχθεί η σωστή εκτέλεση του προγράμματος από τα 4 threads.

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 τις εικόνες που δημιουργήθηκαν από την εκτέλεση του προγράμματος.
- Τα προγράμματά σας (πηγαίοι κώδικες + αναφορά + εικόνες) 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 περίοδο.
- 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.

Προθεσμία παράδοσης: Κυριακή, 11 Μαΐου 2025
Καλή Επιτυχία!
Αναστάσιος Γκαραβέλης