ΕΙΡΗΝΗ ΤΖΙΜΑ P3220201

Άσκηση 1

Μέρος α

Στο πρώτο μέρος της άσκησης μας ζητείται να υλοποιήσουμε την gaussian_blur_separate_parallel(), η οποία παραλληλοποιεί τον κώδικα με την OpenMP. Πιο συγκεκριμένα, χρησιμοποιούμε το #pragma omp parallel for, για να παραλληλοποιήσουμε τα for loops που κάνουν την οριζόντια και την κάθετη θόλωση. Με αυτό τον τρόπο μοιράζονται οι επαναλήψεις σε threads βελτιώνοντας τον χρόνο που χρειάζεται όλη η διαδικασία για να εκτελεστεί.

Πιο αναλυτικά, έχουμε τους παρακάτω χρόνους εκτέλεσης με 4 επαναλήψεις για πιο αξιόπιστα αποτελέσματα:

Επανάληψη	**Serial Time (ms)**	**Parallel Time (ms)**	**Speedup (Serial / Parallel)**
1	2501	436	5.73
2	2225	401	5.55
3	2037	404	5.04
4	2258	434	5.20

Μέσος Χρόνος Serial: (2501 + 2225 + 2037 + 2258) / 4 = 2255.25 ms

Μέσος Χρόνος Parallel: (436 + 401 + 404 + 434) / 4 = 418.75 ms

Μέσο Speedup: 2255.25 / 418.75 =~5.39

Συμπερασματικά λοιπόν, με βάση τα αποτελέσματα παρατηρούμε ότι υπάρχει βελτίωση στον χρόνο εκτέλεσης με την συνάρτηση gaussian_blur_separate_parallel() σε σύγκριση με τη σειριακή gaussian_blur_separate_serial(). Συγκεκριμένα, οι μετρήσεις από τις τέσσερις επαναλήψεις δείχνουν ότι ο μέσος χρόνος εκτέλεσης μειώθηκε από 2255.25 ms (σειριακά) σε 418.75 ms (παράλληλα), κάνοντας έτσι έναν μέσο speedup περίπου 5.39. Η μεγάλη μείωση στον συνολικό χρόνο δείχνει ότι η παραλληλοποίηση βοηθάει

πολύ σε εργασίες όπως η επεξεργασία εικόνας, κάνοντάς τες να ολοκληρώνονται πολύ πιο γρήγορα.

Μέρος β

Στο δεύτερο μέρος της άσκησης, μας ζητείται να υλοποιήσουμε τη συνάρτηση bloom_parallel(), η οποία εφαρμόζει την θόλωσης bloom με τη χρήση του OpenMP για παραλληλοποίηση. Η συνάρτηση εκτελεί τα παρακάτω βήματα:

1. Υπολογισμός μέγιστης φωτεινότητας (luminance): Για κάθε pixel υπολογίζεται η φωτεινότητα του, και στη συνέχεια προσδιορίζεται η μέγιστη τιμή. Η διαδικασία αυτή παραλληλοποιείται με #pragma omp parallel και #pragma omp for nowait, ενώ η ενημέρωση της κοινόχρηστης μεταβλητής max_luminance προστατεύεται με #pragma omp critical ώστε να αποφευχθούν race conditions.

2. Εκτύπωση της μέγιστης τιμής φωτεινότητας**:** Για να εκτυπώσει μόνο ένα νήμα τη μέγιστη τιμή, χρησιμοποιείται η οδηγία #pragma omp single.

3. Δημιουργία της μάσκας bloom: Για κάθε pixel υπολογίζεται αν η φωτεινότητά του ξεπερνά ένα όριο (threshold). Αν ναι, περιλαμβάνεται στη μάσκα. Αυτό το στάδιο παραλληλοποιείται εύκολα με #pragma omp parallel for, καθώς κάθε pixel μπορεί να υπολογιστεί ανεξάρτητα.

4. Εφαρμογή Gaussian blur**:** Εκτελούνται δύο περάσματα (οριζόντιο και κατακόρυφο) με χρήση της συνάρτησης blurAxis. Οι βρόχοι για κάθε πέρασμα παραλληλοποιούνται με #pragma omp parallel for, καθώς κάθε γραμμή/στήλη μπορεί να υπολογιστεί ανεξάρτητα.

5 & 6 & 7. Αποθήκευση θολωμένης εικόνας και δημιουργία τελικής εικόνας: Εφόσον οι δύο αυτές ενέργειες είναι ανεξάρτητες, εκτελούνται παράλληλα με #pragma omp sections, όπου κάθε #pragma omp section χειρίζεται είτε την αποθήκευση της θολωμένης εικόνας είτε τον υπολογισμό και αποθήκευση της τελικής εικόνας. Επιπλέον, στο δεύτερο section παραλληλοποιείται το βρόχος συγχώνευσης των εικόνων με #pragma omp parallel for.