

## Παράλληλος Προγραμματισμός 2020

### Προγραμματιστική Εργασία #1

(Προσοχή: η παράδοση της άσκησης θα γίνει μέσω *github*. Διαβάστε τις οδηγίες στο τέλος της εκφώνησης)

#### Θέμα

Στην επεξεργασία εικόνας συχνά εμφανίζεται ο μετασχηματισμός, κατά τον οποίο η αρχική τιμή κάθε **pixel** αντικαθίσταται από μια νέα τιμή υπολογισμένη από **τα 8 γειτονικά pixels** (και το ίδιο). Κατά τον μετασχηματισμό, κάθε αρχικό pixel πολλαπλασιάζεται με μια σταθερά **float**.

Έτσι, αν έχουμε τα γειτονικά pixels **P0 . . P8** σε διάταξη δύο διαστάσεων:

**P0 P1 P2**

**P3 P4 P5**

**P6 P7 P8**

και τις σταθερές μετασχηματισμού (στην ίδια διάταξη δύο διαστάσεων):

**K0 K1 K2**

**K3 K4 K5**

**K6 K7 K8**

τότε η νέα τιμή για το pixel **P4** ισούται με:

$$\text{newP4} = \text{P0} * \text{K0} + \text{P1} * \text{K1} + \text{P2} * \text{K2} + \text{P3} * \text{K3} + \text{P4} * \text{K4} + \\ + \text{P5} * \text{K5} + \text{P6} * \text{K6} + \text{P7} * \text{K7} + \text{P8} * \text{K8}$$

Η διαδικασία επαναλαμβάνεται για όλα τα pixel, και για όλες τις γραμμές της εικόνας.

Για τις ανάγκες της άσκησης θεωρήστε ότι:

- Η εικόνα είναι **μονόχρωμη** (ένας **float** αριθμός ανά pixel).
- Η εικόνα έχει ήδη μετατραπεί και φορτωθεί σε έναν πίνακα **float NxM** (αρχικοποιήστε με **τυχαίες τιμές** τον πίνακα πριν αρχίσετε τη μέτρηση του χρόνου).
- Οι νέες τιμές που υπολογίζετε θα αποθηκεύονται **σε δεύτερο πίνακα** με τις ίδιες διαστάσεις (αρχικοποιήστε και αυτόν τον πίνακα για να μεταφερθεί στην κρυφή μνήμη πριν αρχίσετε να μετράτε χρόνο).
- Επιλέξτε τα **M** και **N** ώστε να βολεύουν στις πράξεις σας. Μπορείτε π.χ. να θεωρήσετε ότι το N είναι πολλαπλάσιο του 4 (ή οτιδήποτε άλλο σας βολεύει).
- Μπορείτε, αν θέλετε, να αγνοήσετε τον υπολογισμό των pixels στο περίγραμμα της εικόνας.
- Για τις ανάγκες της άσκησης, τα **K0, K1, K2, K3, K5, K6, K7, K8** ισούνται με **0.5**, ενώ το **K4** είναι ίσο με **5.0**.

Ζητείται η επιλογή όσο το δυνατόν αποδοτικότερων SIMD πράξεων **από το σετ SSE2** για την υλοποίηση του μετασχηματισμού.

#### Ζητούμενο

Ο στόχος της άσκησης είναι:

α) Να κατασκευάσετε δοκιμαστικό πρόγραμμα σε C χωρίς εντολές SSE, έτσι ώστε να μελετήσετε τον χρόνο εκτέλεσης χωρίς πρόσθετες βελτιώσεις. Χρησιμοποιήστε πίνακες που χωράνε στην κρυφή μνήμη (L2) του συστήματός σας, έως π.χ. 1000x1000.

β) Κατασκευάστε το αντίστοιχο πρόγραμμα με εντολές SSE. Μετρήστε τους αντίστοιχους χρόνους εκτέλεσης και εισάγετε περαιτέρω βελτιώσεις, αν χρειάζεται. **Βεβαιωθείτε ότι οι αναγνώσεις και εγγραφές είναι ευθυγραμμισμένες στα 16 bytes!**

Δείτε στο site του μαθήματος (<http://mixstef.github.io/courses/parprog/lecture.html>) για παραδείγματα και πρόσθετες πληροφορίες.

### **Παραδοτέο**

Η παράδοση θα γίνει μέσω github. Οδηγίες:

1. Αντιγράψτε (**fork**) το repository **<https://github.com/mixstef/parprog1920a1>** στο δικό σας repository. Βεβαιωθείτε ότι δουλεύετε αποκλειστικά στο **master branch**.
2. Τροποποιήστε κατάλληλα τα αρχεία που περιέχονται στο repository σας με το δικό σας περιεχόμενο:
  - Συμπληρώστε τα στοιχεία σας στο αρχείο **README.md**.
  - Βάλτε τον κώδικά σας στα αρχεία **no-sse.c** και **sse.c** (οι δύο παραλλαγές που εξετάζετε).
  - Προσθέστε την αναφορά σας ως **report.pdf**.
  - **Προσοχή: πρέπει να διατηρήσετε τα ονόματα των παραπάνω αρχείων!**
3. Ενημερώστε το repository σας στο github εντός προθεσμίας. **Μην κάνετε pull request!**

**Η εργασία είναι αυστηρά ατομική.** Για την εγκυρότητα της υποβολής σας θα χρησιμοποιηθεί η χρονοσήμανση των αλλαγών (commits) των αρχείων σας.

**Προθεσμία παράδοσης: 27/3/2020.**