

Παράλληλος Προγραμματισμός Εργασία 1

Διαμαντόπουλος Δημήτρης
Π2016141

Αναφορά

Ο κώδικας αφού αρχικοποιήσει 2 πίνακες με τυχαίες float τιμές υπολογίζει τα pixel μιας εικόνας(array) με βάση κάποιες σταθερές. Το περίγραμμα αγνοείται, ενώ όλα τα εσωτερικά pixels, δηλαδή από το δεύτερο pixel της δεύτερης γραμμής μέχρι και το προτελευταίο pixel της προτελευταίας γραμμής υπολογίζονται ως εξής:

```
for(int i = 1; i < N-1; i++){ //trexei apo thn deuterh grammh mexri thn proteleutaia
    int px=0;
    for(int j = 1; j < M-1; j++){ //trexei apo to deuthero pixel kathe grammhs grammh mexri to proteleutaio
        px++;
        final_img[i * M + px] = img[(i * M + px)-M-1] * k0 + img[(i * M + px)-M] * k1 + img[(i * M + px)-M+1] * k2 +
        img[(i * M + px)-1] * k3 + img[(i * M + px)] * k4 + img[(i * M + px)+1] * k5 + img[(i * M + px)+M-1] * k6 +
        img[(i * M + px)+M] * k7 + img[(i * M + px)+M+1] * k8;
    }
}
```

Αν π.χ το pixel στο οποίο βρίσκομαι τώρα σε έναν πίνακα 200x200 είναι το img[201] δηλαδή το δεύτερο pixel της δεύτερης σειράς, τότε:

το νέο final_img[201]= img[0] * k0 +
img[1] * k1 +
img[2] * k2 +
img[200] * k3 +
img[201] * k4 +
img[202] * k5 +
img[400] * k6 +
img[401] * k7 +
img[402] * k8 +

Για πίνακα 200x200 τα αποτελέσματα του πρώτου, ενός τυχαίου και του τελευταίου pixel είναι:

```
Array size: 40000  
  
img[201]: 95.474998(first pixel that can be calculated)  
final_img[201]: 859.274963  
  
img[401]: 190.475006(random pixel)  
final_img[401]: 1714.275024  
  
img[39798]: 18904.050781(last pixel that can be calculated)  
final_img[39798]: 170136.453125
```

Ενώ ο χρόνος είναι:

```
Time: 0.000101  
  
MFLOPS/sec = 791.378113
```

Το κομμάτι sse.c δεν υλοποιήθηκε πλήρως.