Documentație Laborator 2

Student: Rentea Dan

Grupa: 236/2

## Analiza cerințelor

Se cere aplicarea unui filtru peste o imagine reprezentata cu o matrice. Aplicarea filtrului trebuie sa fie făcută si liniar si paralel si sa comparam timpii de execuție.

## Proiectare

Am creat o clasa Matrice. Aceasta reprezintă numerele cu un vector de vectori.

Pentru filtrarea liniara se parcurge matricea pe linii de sus in jos si se aplica filtrul pe fiecare celula.

Pentru aplicarea filtrului se parcurge blocul de pixeli in care cel curent este centrul. Daca se încearcă accesarea unui pixel out of bounds se calculează ca si 0. Deoarece schimbările se fac direct in matricea curenta, fiecare thread pre-parcurge matricea inițială si tine minte elementele de care va avea nevoie. Pentru sincronizarea intre thread-uri in Java folosesc un CyclicBarrier, iar in C++ un semafor custom pe baza de mutex si variabile condiționale.

Pentru filtrarea paralela se împart elementele in mod egal intre thread-uri. Se considera ca matricea ar fi de fapt un vector continuu, se împart elementele pe chunks, iar după aceea thread-ul transforma din poziția numărului in vector in poziția din matrice. In acest mod ne putem asigura ca numărul de operații este împărțit in mod egal intre thread-uri.

## Tabela cu timpii de execuție

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Durata(nanosecunde) Java | Durata (nanosecunde) C++ | Nr. Threads | Linii | Coloane |
| 204020 | 3320 | 0 | 10 | 10 |
| 4717760 | 4113040 | 4 | 10 | 10 |
| 56514780 | 80688140 | 0 | 1000 | 1000 |
| 1172472200 | 28468020 | 2 | 1000 | 1000 |
| 648838540 | 27115260 | 4 | 1000 | 1000 |
| 480491320 | 26692420 | 8 | 1000 | 1000 |
| 458541060 | 23108960 | 16 | 1000 | 1000 |
| 4126760 | 6183100 | 0 | 10 | 10000 |
| 102217960 | 4095320 | 2 | 10 | 10000 |
| 62448640 | 4010180 | 4 | 10 | 10000 |
| 38976420 | 4078400 | 8 | 10 | 10000 |
| 37006860 | 4072140 | 16 | 10 | 10000 |
| 4826040 | 4578060 | 0 | 10000 | 10 |
| 76696180 | 4350840 | 2 | 10000 | 10 |
| 43021520 | 4687542 | 4 | 10000 | 10 |
| 29406600 | 5054180 | 8 | 10000 | 10 |
| 27690600 | 4646020 | 16 | 10000 | 10 |

## Diagrama de clase

