

# Tema 1

22 December 2023 00:27

Nume: Bontaș Cezar-Octavian

Grupa: 332AC

În cadrul fisierului "process.v" am rezolvat cerințele MIRROR și GRAYSCALE, FILTER având descrisă în cadrul acestui README o propunere de implementare. O să descriu puțin pe rând ce face fiecare subrutină:

- MIRROR:
  - Are 5 stări componente (3 propriu-zis conțin algoritmul cerinței, MIRROR\_INIT și MIRROR\_FINISH sunt doar pentru inițializare / încheiere rutină);
  - Parcurgerea matricei este făcută coloană cu coloană, pașii fiind simetrici față de axa orizontală între liniile 31 și 32;
  - Ordine parcurgere: 0 -> 63 -> 1 -> 62 -> 2 -> 61 -> ... -> 31 -> 32 -> se termină de parcurs elementele de pe coloana la care ne aflăm;
  - Practic MIRROR se oprește atunci când ajunge la pixelul de pe poziția (31, 63);
  - MIRROR\_STEP1 se ocupă de partea superioară a matricei, MIRROR\_STEP2 de cea inferioară, MIRROR\_STEP3 revine în partea de sus, și ia decizia de a continua rutina dacă mai are ce parcurge din imagine.
- GRAYSCALE:
  - Aici este relativ simplu, parcurgerea se face linie cu linie, pe fiecare element din linie se face media. Nu cred că necesită explicații adiționale.
- FILTER:
  - Nu am avut timp să scriu cod practic pentru sharpness, dar am o idee legată de o soluție. Nu e neapărat de punctaj, o fac doar pentru completarea documentului.
  - Pași algoritm filtrare:
    1. Inițializare cache matrice. În implementarea gândită, avem nevoie doar de 3 linii. Cache-ul o să fie puțin mai lung decât lungimea liniei, ca să putem să punem valorile de 0 preventiv pentru marginile matricei. Pentru pasul ăsta, doar 2 linii trebuie parcurse ca să fie citite (linia 0 și 1).
    2. Aplicăm filtrul de sharpness, având în vedere că toate valorile utile sunt cacheuite, pe o linie.
    3. Citim linia care urmează (dacă există, dacă nu, adică ultima linie, atunci se umple cea mai nouă linie din cache cu 0-uri) și memorăm valorile în cache.
    4. Permutăm liniile de la linia cea mai nouă la cea mai veche, cu precizarea că cea mai veche linie nu mai e utilă, și atunci e suprascrisă cu linia 2.
  - Asta este în teorie, în practică nu am implementat încă și probabil mai bine las așa decât să îmi complic viața și să risc să pierd punctele că nu predau la soft deadline.

În cod sunt puse și comentarii pentru explicarea unor părți de cod care nu pot fi clare (deși în README e scrisă documentația, comentariile din cod sunt doar pentru siguranță) sau marchează porțiuni de cod (e.g: rutinele).