**Proiect 2 (CUDA)**

Salsigan Razvan-Dan

Sabo-Brandus Alina

Grupa 236/2

Data predare: 11.01.2022

**Cerinta aleasa**

Aplicarea filtrului Laplacian asupra imaginilor, folosind CUDA

**Implementare**

Imaginea trebuie sa fie un patrat!

Se citeste imaginea folosind OpenCV in format RGBA, dupa care se extrag pixelii in matrici liniarizate (vectori) de tipul uchar4.

1. Pentru filtrul cu kernelul

**1 1 1**

**1 8 1**

**1 1 1**

Se extrage fiecare culoare intr-o matrice liniarizata, dupa care se aplica filtrul asupra fiecareia dintre ele. La final se refac cele 3 culori intr-un pixel rgb.

1. Pentru filtrul cu kernelul

**0 -1 0**

**-1 4 -1**

**0 -1 0**

Se converteste matricea de pixeli rgb la una grayscale, dupa formula

grayscale[i] = 0.2989 \* rgba.x + 0.5870 \* rgba.y + 0.1140 \* rgba.z;

dupa care se aplica filtrul

**Metode folosite**

Metoda care se ocupa de schimbul de date intre placa video (device) si host (CPU)

cudaError\_t filterWithCuda(unsigned char\* matrix, unsigned char\* result\_matrix, float\* filter, int rows, int columns)

Metoda kernel in care se aplica convolutia propriu-zisa. Se calculeaza indiciele liniei si coloanei corespunzatoare elementului pentru care se apeleaza,

matricile fiind liniarizate. Se verifica daca exista depasiri si se aplica kernelul asupra pixelului curent.

\_\_global\_\_ void filterKernel(unsigned char\* matrix, unsigned char\* result, float\* filter, int rows, int columns, int filter\_rows, int filter\_columns)

**Rezultate**

|  |  |  |  |
| --- | --- | --- | --- |
| **1400x1400 Original** | **Filtrul 1.** | **Filtrul 2.** | **Filtrul 1 + 2** |
|  |  |  |  |
|  |  |  |  |