

# Procesare de imagini- Operatorul Laplace

Stanciuлесcu Andrei – 333AA

## Introducere

In procesarea de imagini, detectarea marginilor este un element esential pentru a putea dezvolta algoritmi de nivel mai inalt. Pentru a realiza acest lucru, in general algoritmi cauta zone in care apare o schimbare rapida a nivelului de culoare sau a luminozitatii pixelilor. Astfel de zone pot reprezenta tranzitia intre doua obiecte din imagine sau o tranzitie obiect fundal. Pentru a analiza in acest fel imaginile, putem folosi gradientul imaginii, adica viteza de variatie a valorilor pixelilor pe o anumita axa.

## Prezentarea teoretica si descrierea implementarii

Operatorul Laplace calculeaza a doua derivata a imaginii in fiecare punct pe cele doua directii X si Y. Intrucat imaginile stocate sunt formate dintr-un numar finit de pixeli, trebuie aplicata o varianta discretizata a operatorului Laplace. Acest lucru poate fi facut prin convolutia 2D intre imaginea initiala si un nucleu de convolutie ales corespunzator. In aceasta aplicatie vom folosi un nucleu de dimensiune 3x3 de forma :

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Acest nucleu produce operatia Laplace negativa, deoarece in zona de tranzitie va avea valori negative in parte in care imaginea este mai luminoasa.

Operatia de convolutie se poate face individual pe fiecare canal de culoare al imaginilor, insa acest lucru nu este folositor in general. Astfel, vom transforma mai intai imaginea din format RGB in Grayscale, apoi vom aplica operatia de convolutie pe imaginea rezultata ce are un singur canal si ofera un rezultat mai relevant, intrucat zonele de variatie a luminozitatii indica tranzitia intre obiecte ale imaginii.

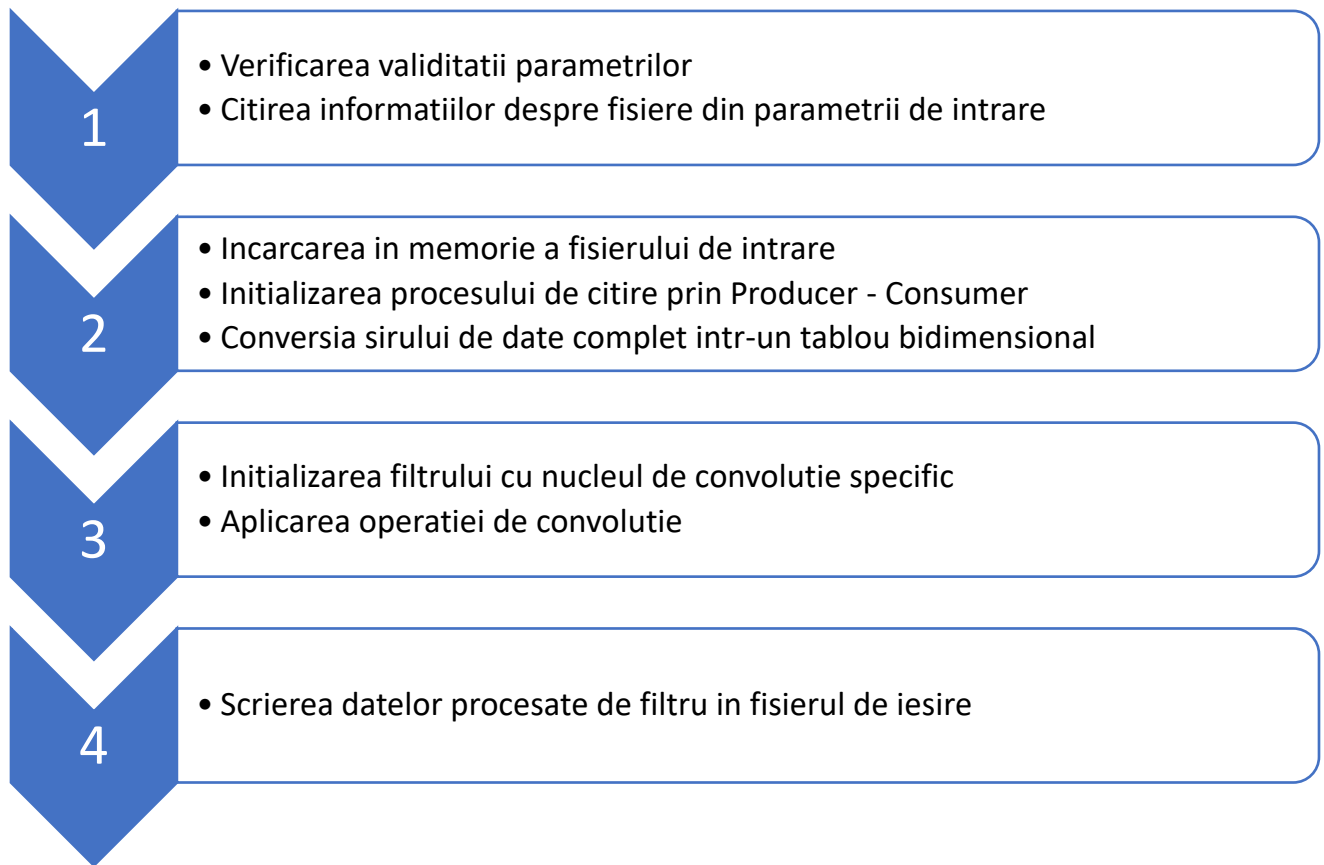
Conversia RGB – Grayscale se face printr-o metoda ponderata ce tine cont de perceptia ochiului uman a luminozitatii culorilor astfel :

$$G = 0.3 * R + 0.59 * G + 0.11 * B$$

Mai mult de atat, raspunsul filtrului este pozitiv in unele zone ale imaginii si negativ in altele, iar valorile negative nu pot fi reprezentate corect in formatul BMP. Pentru a rezolva aceasta problema, vom aduna la iesirea filtrului un offset de 122 (valoarea este la jumatatea intervalului posibil de valori). In acest fel vom putea observa atat raspunsul pozitiv cat si cel negativ al filtrului pe un fundal constant.

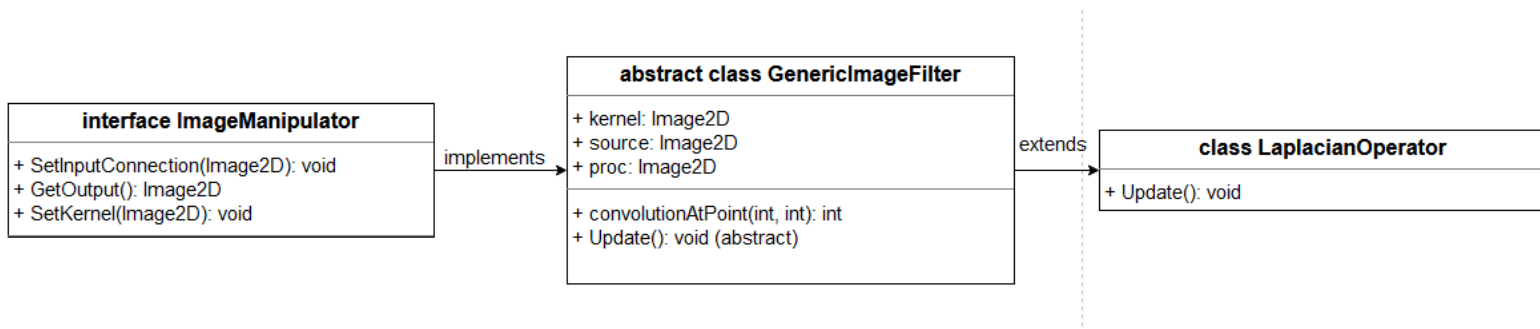
Pentru a calcula raspunsul filtrului pe o anumita imagine, se va folosi operatia de convolutie 2D, ce presupune suprapunerea filtrului pe fiecare pixel al imaginii initiale si se calculeaza suma produselor dintre valorile filtrului si valorile corespunzatoare in imagini. In cazurile marginilor, portiuni din filtru depasesc limitele imaginii. Aceasta problema poate fi rezolvata prin extinderea imaginii folosind o tehnica de bordering, insa nu acesta era scopul acestei lucrari. Astfel, in cazurile de margine, cand filtrul nu se suprapune perfect, vom considera raspunsul 0.

## Etapele de executie



## Ierarhia obiectelor de filtrare a imaginii

Pentru a avea o implementare modulara a secventei de procesare a imaginii, structura filtrului a fost implementata pe mai multe nivele. Se incepe cu o interfata generala pentru obiectele de lucreaza cu imagini, care include metodele pentru selectarea imaginii de intrare, a nucleului si pentru extragerea rezultatului. Interfata e implementata de un obiect ce defineste un filtru general de convolutie. Acesta introduce campurile in care vor fi stocate datele despre imagini si implementeaza convolutia intr-un punct. In final, operatorul Laplace extinde aceasta clasa abstracta, implementand concret metoda Update si folosind concret nucleul specific acestei operatii.



## Descrierea modulelor

### Image2D

Aceasta clasa este folosita pentru a stoca date despre imagini, precum inaltimea si latimea acesteia si valorile pixelilor intr-un tablou bidimensional. Acesta permite modofocarea unei singure valori sau a intregului tablou.

### ImageProducer

Aceasta clasa preia datele din imaginea de intrare, le interpreteaza ca format RGB, face conversia catre grayscale, apoi pune noile date intr-un buffer pentru a fi luate de Consumer.

**OBS:** Numarul de pixeli din imagine trebuie sa fie multiplu de 4 pentru a putea imparti imaginea in sferturi asa cum este specificat in cerinta.

### ImageConsumer

Aceasta clasa preia datele din buffer si le aranjeaza in sirul de date final.

### ImageBuffer

Acesta este buffer-ul intermediar ce mediaza comunicarea intre Producer si Consumer

### ImageReader

Aceasta clasa se ocupa de deschiderea imaginii de intrare. Apoi, lanseaza in executie in thread-uri separate Producer si Consumer. Cand acestea si-au finalizat executia, datele sunt preluate din consumer si aranjate intr-un tablou bidimensional, apoi introduse intr-un obiect Image2D.

### ImageWriter

Aceasta clasa preia datele dintr-un obiect Image2D si le scrie pe rand, in ordinea corecta, in fisierul de iesire.

Functionarea modulelor de procesare a imaginii au fost explicate in primele 2 capitole ale documentului

## Rezultate

Am rulat aplicatia pe mai multe date de intrare, pentru a evidentia rezultatul in diferite conditii:



Figure 1: 1323x740, 50ms

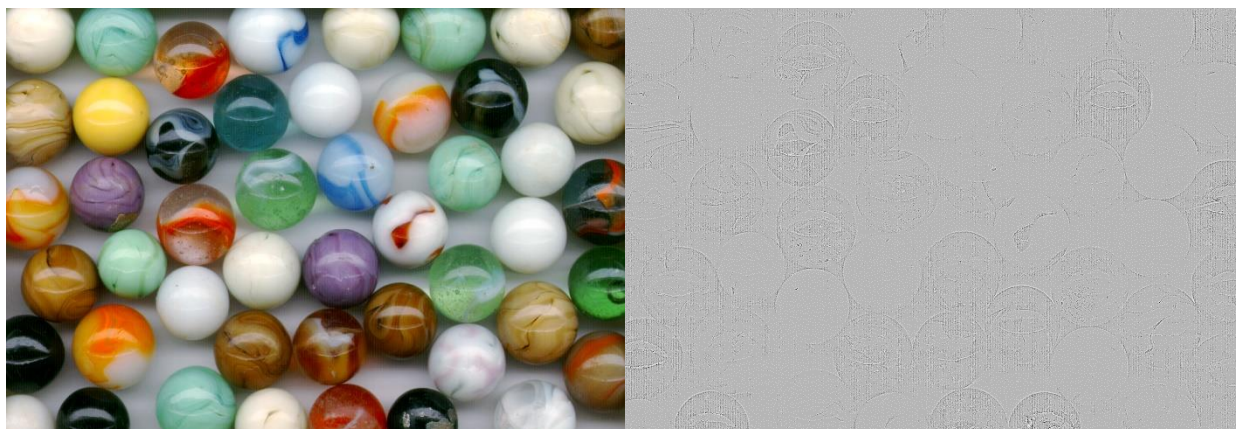


Figure 2: 1390x948, 62ms

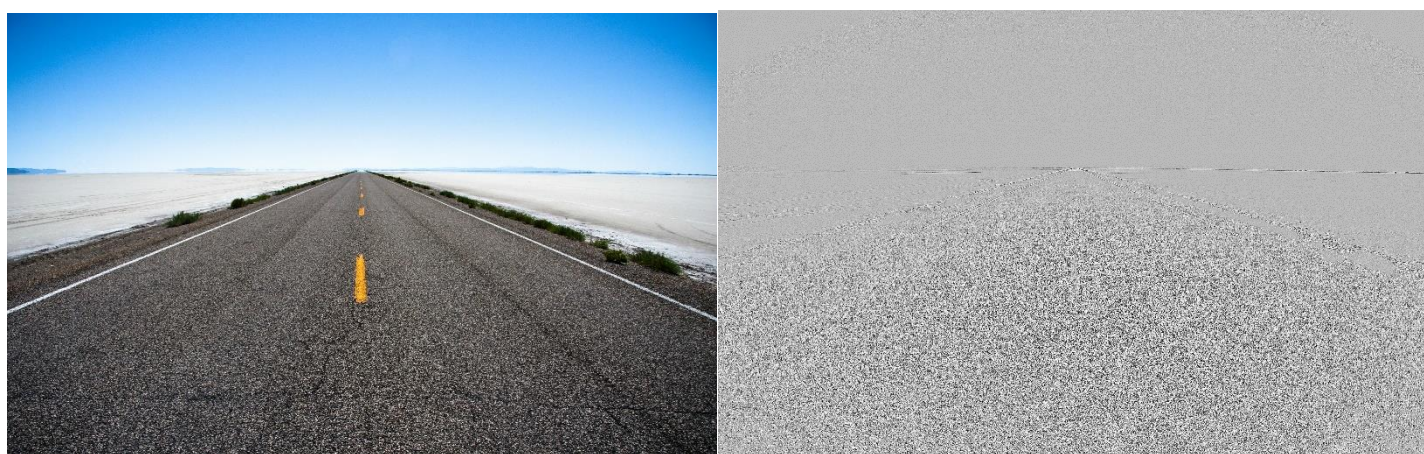


Figure 3: 3396x2123, 300ms

## Concluzii

Observam ca in exemplele in care obiectele din imagine au culori uniforme, algoritmul reuseste sa puna in evidenta interfata dintre obiecte sau dintre obiect si fundal. Cu toate acestea, Operatorul Laplace este foarte sensibil la zgomot, asa cum se poate observa in a treia figura. Culoarea asfaltului nu este uniforma, iar acele variatii rapide in intensitate duc la un raspuns exagerat al filtrului. Pentru a compensa pentru aceste neajunsuri, se poate folosi un filtru denumit Laplace of Gaussian (LoG). Acesta implementeaza, pe langa componenta operatorului Laplace, un filtru de mediere gaussiane ce are ca rol eliminarea zgomotului din imagini si generarea unor tranzitii mai uniforme in intensitate. Pentru a folosi un astfel de filtru este nevoie de un nucleu de convolutie de dimensiune semnificativ mai mare, pentru a avea efect fenomenul de mediere.

## Bibliografie

1. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/>
2. <https://docs.oracle.com/javase/tutorial/2d/>