Student: Pantu Mihnea-Andrei 331 AB

Profesor coordonator: prof.dr.ing Andrei Hossu

Image resizing

Zero Order Hold Method

**Introducere**

Prin intermediul aplicatiei create avem posibilitatea de a mari o imagine in format bmp de doua ori. Rezultatul procesului va fi stocat intr-un nou fisier ce va putea fi accesat de utilizator.

**Descrierea aplicatiei cerute**

Aplicatia noastra ii ofera utilizatorului optiunea de a introduce in terminalul ei numele imaginii pe care vrea sa o mareasca si numele imaginii prelucrate. Inainte de asta utilizatorul trebuie sa introduca calea de fisiere in care se regaseste poza. Ulterior, aplicand asupra acesteia meotda Zero Order Hold, aplicatia mareste imaginea si o salveaza intr-un nou fisier. De asemenea, intr-un alt fisier va fi salvata o copie a imaginii cu dimensiunile initiale in caz ca utilizatorul vrea sa revada cum arata imaginea inainte de procesul de zoom.

\*menituni

Numele pozelor trebuie introdus fara extensie.

Pentru introducerea caii de fisiere, exemplu: C:\\Users\\pantu\\downloads\\.

**Notiuni teoretice:**

Metoda Zero Order Hold este o metoda ce mareste o imagine, cunoscuta si cu numele de Zoom Twice (marire de 2 ori), deoarece prin aceasta metoda, o imagine poate fi marita de doar 2 ori.

In cadrul acestei metode, se aleg doua elemente adiacente de pe linii, le adunam apoi impartim rezultatul la 2, aproximam la un intreg rezultatul obtinut si il plasam intre cele 2 elemente. Se aplica acest procedeu intai pe linii apoi pe coloane.

Ne-am folosit de valorile pixelilor imaginii initiale pe care le-am stocat intr-o matrice si apoi am amplicat conceptul metodei.

Exemplu:

Vom lua o imagine de 2x2 si o vom mari de 2 ori folosind metoda Zero Order Hold:

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

Prima data vom aplica procedeul pe linii apoi pe coloane.

1+2 = 3; 3/2 = 1.5 ~ 1, respectiv 3+4 = 7; 7/2 = 3.5 ~3

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 2 |
| 3 | 3 | 4 |

Apoi aplicam pe coloane:

1+3 = 4; 4/2 = 2, respectiv 2+4 = 6; 6/2 = 3

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 2 |
| 2 | 2 | 3 |
| 3 | 3 | 4 |

Imaginea initiala avea dimensiunea 2x2, cea finala in urma aplicarii metodei Zero Order Hold are dimensiunea 3x3. Aplicand metoda inductiei se ajunge la rezultatul urmator:

O imagine care trece prin procesul de prelucrare Order Hold Method:

(2 \* Nr de Linii - 1) x (2 \* (Nr de Coloane) – 1).

Avantaje:

Metoda are ca rezultat imaginea marita de 2 ori, iar rezultatul final este clar comparativ cu rezultatul altor metode precum metoda Nearest Neighbor Interpolation.

Dezavantaje:

Metoda Zero Order Hold poate mari doar de doua ori imaginea initiala.

**Descrierea structurala, arhitecturala si functionala:**

Aplicatia creata este implemntata pe baza conceptelor de programare orientata pe obiecte. Am implementat 8 clase in cadrul carora am integrat toate cele 4 concepte ale OOP: incapsulare, mostenire, polimorfism si abstractizare.

Clasele utilizate:

Image, in care am implementat metoda Zero Order Hold

Infos, mostenita din Image, contine metode de afisare a denumirii modalitatii de zoom utilizate.

Dimensions, mostenita din Infos, foloseste varargs si stabileste dimensiuni maxime admise pe care le poate avea o imagine ce urmeaza a fi prelucrata si le afiseaza utilizatorului in terminal.

InterfataAbstracta, clasa abstracta cu metode abstracte.

Main care mosteneste InterfataAbstracta, in care am implementat metodele din clasa abstracta, concatenand informatii cu privire la creatorul aplicatiei.

Threads, implementate in clasele Consumer, Producer si Buffer.

**Descrierea implementarii:**

In clasa Image, variabilele width si height pastreaza dimensiunile imaginii initiale, iar newWidth si newHeight sunt dimensiunile noii metode. Am implementat constructor pentru initializarea parametrilor. Am implementat setters si getters pentru a seta si accesa valorile acestor variabile. Avem si variabilele private nrLin si nrCol private care afiseaza nr de linii si coloane ale matricei in care vom stoca valorile pixelilor imaginii initiale. Au implementate metode de setters si getters, astfel se utilizeaza procesul de incapsulare. Am utilizat metodele respective si in sublcase. Am creat metode denumite la fel, dar cu parametrii diferiti, folosindu-ma de polimorfism pentru a afisa diverse aspecte legate de dimensiunile matricei sau de metoda pe care o folosesc in procesul de zoom al imaginii.

Am implementat o functie de citire a imaginii ce urmeaza sa fie marita, apoi in fucntia main am inceput implementarea metodei Zero Order Hold. Am oferit utilizatorului optiunea de a introduce de la tastatura denumirea fisierului ce urmeaza sa fie prelucrat, ulterior folosim metodele anterioare pentru a obtine dimensiunile imaginii initiale apoi parcurgem imaginea pixel cu pixel, obtinem valorile pixelilor si le stocam intr-o matrice auxiliara.

Ulterior initializam matricea in care vom stoca valorile finale ale pixelilor imaginii marite si punem in aplicare metoda Zero Order Hold pe linii si pe coloane si setam pixelii imaginii finale.

Pentru a lucra cu valorile pixelilor am folosit functiile getAlpha(), getRed(), getBlue(), getGreen() si setRGB.

In final, vom scrie in 2 fisiere noi imaginea rezultata in urma prelucrarii si imaginea initiala fara zoom. Ele vor putea fi vizualizate deschizand fisierele cu denumirea oferita de utilizator in cazul imaginii marite si cu denumirea „zoomout.png” pentru imaginea fara zoom.

Clasa Infos mosteneste clasa Image, foloseste getters si setters pentru a seta si accesa denumirea metodei folosite, denumire ce va fi afisata din main-ul clasei.

Clasa Dimensions mosteneste la randul ei Infos, obtinand astfel o mostenire pe 3 niveluri si foloseste varargs si metode din superclase pentru a afisa utilizatorului dimensiunile maxime admise ale unui imagini.

Clasa InterfataAbstracta este o clasa abstracta cu metoda abstracta ce urmeaza a fi concretizata in clasa Main.

Clasa Main foloseste setters si getters pentru a obtine si accesa numele, prenumele si disciplina creatorului aplicatiei, apoi implementeaza metoda din clasa abstracta si se foloseste de ea pentru a concatena si afisa detaliile despre creator.

Pentru implementarea Threadurilor am creat clasele Producer, Consumer si Buffer.

Descrieri mai specifice fiecarei metode si variabile utilizate sunt mentionate in cadrul portiunilor comentate din codul implementat.

**Concluzii:**

Metoda Zero Order Hold este o metoda relativ usor de implementat ce se foloseste de valorile pixelilor imaginii initiale pentru a o mari de 2 ori, prin preluarea pixelilor adiacenti, calcularea mediei aritmetice a valorilor lor si inserarea aceste medii intre cei doi pixeli, intai pe linii apoi pe coloane. Dezavantajul este insa ca imaginea nu poate fi marita mai mult de 2 ori.

Imaginea inainte de prelucrare:

A screenshot of a video game

Description automatically generated

Imaginea dupa prelucrare:

A screenshot of a computer

Description automatically generated with low confidence

**BIBLIOGRAFIE**

https://www.baeldung.com/java-varargs

[https://www.tutorialspoint.com/dip/Zooming\_Methods.htm#](https://www.tutorialspoint.com/dip/Zooming_Methods.htm)

<https://www.javatpoint.com/thread-sleep-in-java>

<https://docs.oracle.com/javase/7/docs/api/java/awt/image/BufferedImage.html>

<https://github.com/yusufshakeel/Java-Image-Processing-Project>