

Sobel Operator

Descrierea aplicatiei

Aplicatia realizata aplica operatorul Sobel sau filtru Sobel pentru o imagine color cu extensia .bmp pe 24 biti . Imaginea .bmp este formata din o matrice de pixeli , iar valoare unui pixel este de 24 de biti . Pentru culorile Red(Rosu) , Green (Verde) ,Blue (Albastru) se aloca 8 biti din cei 24 biti astfel incat un pixel are forma Pixel(255,255,255) pentru un pixel de culoare alba si pixel(0,0,0) pentru culoare neagra. Restul culorilor se obtin din combinatia celor 3.

Pentru a aplica filtrul Sobel este necesar ca imaginea sa fie in alb-negru . Aplicatia transforma imaginea color in imagine alb-negru media aritmetica a culorilor Rosu,Verde si Albastru. Dupa care pe imaginea nou formata se aplica filtrul sobel.

Filtrul Sobel este un produs de convolutie dintre imaginea in alb-negru si un nucleu de 3x3 care este fixat. In urma produsului de convolutie dintre imagine si nucleu o valoare pe directia $G_x =$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \text{ pe directia } G_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A.$$

Odata ce se obtin aceste valori pentru G_x si G_y trebuie sa extragem radical din suma lui G_x si G_y la patrat

$$G = \sqrt{G_x^2 + G_y^2}$$

Daca valoarea lui G este mare atunci inseamna ca acel pixel face parte din o margine a unui obiect.

Descrierea Implementarii

Aplicatia a fost realizata in Java folosind IDE Eclipse. Aplicatia urmeaza urmatoarele etape : Citeste imaginea din un fisier ,se transforma imaginea citita din fisier in alb-negru , se aplica filtrul Sobel pe imaginea alb-negru , se salveaza noua imagine obtinuta.

Aplicatia foloseste args[0] si args[1] prin care se transmit calea catre fisierul sursa si calea fisierului nou creat.

Imaginea este citita si stocata cu ajutorul Clasei din java.awt.image, BufferedImage ,care este specifica operatiilor cu imagini. Aplicatia consta in 2 etape pentru a prelua imaginea :

Prima etapa care incarca imaginea pe blocuri de 1/4 din dimensiunile imaginii si in acelasi timp blocul respectiv este transformat din color in alb-negru.

A doua etapa este dupa ce s-a terminat de transformat imaginea in alb-negru se aplica metoda *procesare_filtru* care implementeaza filtru Sobel

Procesarea imaginii este implementata multithread folosind metoda Producer-Consumer . Functionand astfel:

Se va crea un Producer thread care va citi cate 1/4 din imaginea din buffer si blocul de 1/4 din imagine color in imagine alb-negru ,dupa ce se termina de procesat 1/4 din imagine se scrie in buffer cu metoda `setPixels` iar threadul producer intra in not Runnable.

Threadul Consumer consuma blocul de imagine procesat de producer din buffer folosind `getPixels` si pune imaginea procesata intr-o imagine noua care va forma imaginea alb-negru finala. Dupa terminare threadul intra in not Runnable.

Pentru sincronizarea si comunicarea dintre threaduri s-au folosit `notify()` si `join()`.

Dupa ce s-a terminat de format noua imagine alb-negru se apeleaza metoda `procesare_filtru()` care aplica filtru Sobel .Pentru fiecare pixel se aplica nucleu Gx si Gy prin convolutie se calculeaza magnitudinea pixelului ca radical suma patratelor Gx si Gy . Se cauta magnitudinea maxima care va fi folosita la scalarea valorii pixelilor pentru a evidentia marginile obiectelor. Se scaleaza gradientul pentru a nu depasi valoare maxima de 255 si se reconstruieste imaginea in care se evidentiaza marginile obiectelor.

Se salveaza imaginea in calea data in `args[1]`.

Evaluarea performantelor

Timpul de executie al programului este influentat de dimensiunea imaginii ,cum imaginile pot avea dimensiuni variate acestea dimensiunile acestora variaza timpul de executie al programului. Pentru a cronometra timpul necesar de completare al etapelor folosim metoda `System.nanoTime()` care sa salveze timpul de incepere si terminare al etapelor. Vom face diferenta dintre cele doua pentru a obtine timpul necesar completarii etapei . Am folosit 2 imagini una cu rezolutie de 3840 pe 2560 de pixeli si am obtinut:

```
Timpul pentru citire: 0.0719715 secunde  
Timpul pentru procesare: 7.2671281 secunde  
Timpul pentru salvare: 0.1000456 secunde
```

Din timp pentru procesare trebuie sa se scada 4 secunde deoarece sunt puse de noi in cod, deci rezulta un timp de 3,2 secunde de procesare a imaginii

Pentru o imagine de 997 pe 559 dureaza 0,2 secunde .

```
Timpul pentru citire: 0.0605111 secunde  
Timpul pentru procesare: 4.1940473 secunde  
Timpul pentru salvare: 0.0292749 secunde
```

Se observa o crestere a timpului de procesare odata cu cresterea rezolutia imaginii.

Concluzie

Filtrul Sobel extrage din imaginea initiala marginile obiectelor facandu-le culori deschise iar restul culori spre negru. Pentru a putea aplica filtru Sobel trebuie sa transformam imaginea color in alb-negru . Dupa ce imaginea este alb-negru putem aplica filtrul Sobel pentru a evidentia marginile obiectelor din imagine.

Giurgea George
332AB

Bibliografie

https://en.wikipedia.org/wiki/Sobel_operator

https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm

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