



Operatorul Sobel

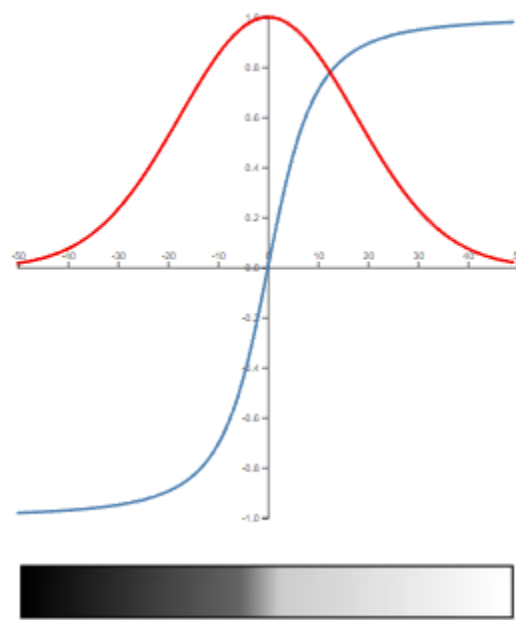
SIMA MARIA-CRISTINA

UPB, ACS – IS, GRUPA 331AB

Descriere matematică a operatorului Sobel

Operatorul Sobel este utilizat în procesarea de imagini, în mod particular pentru algoritmi de edge detection. Acesta creează o imagine alb-negru, în care sunt evidențiate marginile.

Din punct de vedere matematic, o schimbare bruscă a unei curbe pe un grafic poate fi determinată prin panta ei, adică prima derivată la fiecare punct de-a lungul curbei. Ca în graficul de mai jos (unde linia albastră este inia intensității imaginii, iar cea portocalie este linia pantei), se observă prezența unei margini în mijlocul graficului (deci acolo este o arie unde intensitatea imaginii se schimbă brusc).



Operatorul Sobel face convoluția între imagine și două matrici de 3x3, una pentru schimbările orizontale și una pentru cele verticale.

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G}_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * ([-1 \quad 0 \quad +1] * \mathbf{A}) \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} * ([1 \quad 2 \quad 1] * \mathbf{A})$$

Kernel-ul va fi elementul din mijlocul matricii de 3x3, iar fiecare celulă din împrejurul acestuia se va înmulți cu corespondentul din matricea noastră A. De exemplu,

Selection			*	Kernel			=	Result			Change:
1	1	100		-1	-2	-1		-1	-2	-100	447
1	100	150		0	0	0		0	0	0	
100	150	150		1	2	1		100	300	150	

După aceea, gradientul (aproximarea) se poate calcula utilizând formula (care ne și ajută să scăpăm de semn):

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

Descrierea programului

Programul aplică operatorul Sobel pe o imagine introdusă de la tastatură.

Aplicația are 2 package-uri, unul pentru aplicația de test (main), iar celălalt pentru restul claselor.

Restul claselor sunt: ImageAbstractă (clasă abstractă, utilizată pentru crearea interfeței), InterfataImage, ImageColor (derivată din ImageAbstracta) și ImageAlbNegru (derivată din ImageColor).

Proiectul conține multe comentarii utile, pentru o înțelegere ușoară a modalității de rezolvare alese, și conține principiile programării orientate pe obiecte (incapsulare, moștenire, polimorfism, abstractizare). Aplicația lucrează cu fișiere, este multimodulară și tratează excepții.

Evaluare performanțe



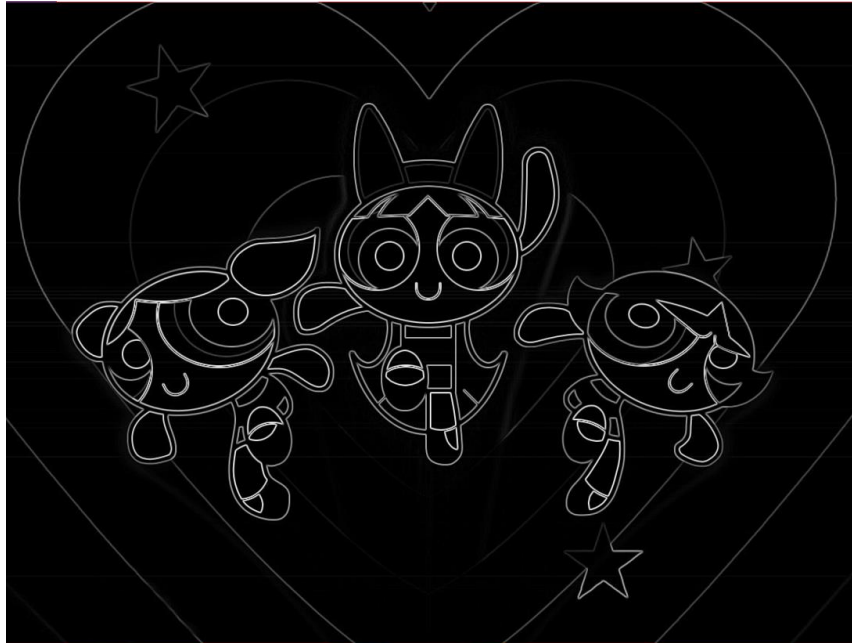
Pentru poza "powerpuff.bmp", pentru aplicarea operatorului Sobel fara a realiza Greyscale anterior, obtinem urmatoarele rezultate:

```
Algoritmul Sobel a durat: 1.42606336E8 milisecunde  
Salvarea Imaginii dupa aplicarea operatorului Sobel a durat: 746.5861 milisecunde  
! Crearea pozei 1 a durat: 749 milisecunde.
```



Pentru aplicarea Greyscale-ului prima data, iar dupa aceea aplicarea operatorului Sobel, s-au obtinut urmatoarele rezultate:

```
Algoritmul Sobel Grey a durat: 125.82912 nanosecunde  
Salvarea sobel grey a durat: 813.69495 nanosecunde  
! Crearea pozei 2 a durat: 818 milisecunde.
```



Concluzii

Asadar, se observa faptul ca a doua varianta, in care aplicam intai Greyscale, este mai costisitoare din punctul de vedere al executiei aplicatiei, dar obtinem rezultate mai precise decat in primul caz, cand aplicam operatorul in mod direct.

Bibliografie

1. <https://stackoverflow.com/questions/41468661/sobel-edge-detecting-program-in-java>
2. <https://automaticaddison.com/how-the-sobel-operator-works/>
3. <https://cse442-17f.github.io/Sobel-Laplacian-and-Canny-Edge-Detection-Algorithms/>
4. https://en.wikipedia.org/wiki/Sobel_operator
5. <https://www.youtube.com/watch?v=uihBwtPIBxM>