

Găsirea Contururilor În Imaginile Digitale

Tache Daria Elena Zavoianu Catinca

Facultatea de Matematică și Informatică
Universitatea din București
Specializarea Informatică

January 31, 2025

Cuprins

Introducere

Metode Matematice

Operatori de Gradient

Operatori Gaussieni

Concluzii și Comparații

Biblioteci Folosite

Bibliografie

Introducere

Proiectul despre detecția marginilor se concentrează pe identificarea marginilor în imagini digitale, care sunt definite ca linii sau curbe unde luminozitatea imaginii suferă schimbări bruste sau întreruperi.

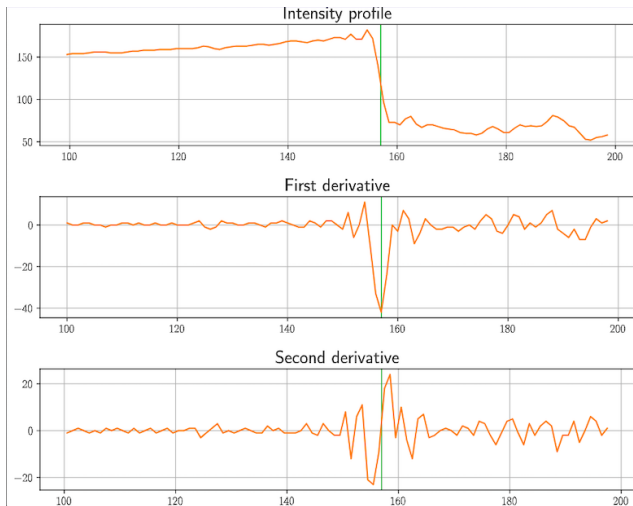
- ▶ Detecția marginilor este esențială în procesarea imaginilor.
- ▶ Aplicații: recunoașterea obiectelor, realitatea augmentată, imagistica medicală.
- ▶ Probleme practice: fragmentarea marginilor, margini false, segmente lipsă.

Derivate și Margini

- ▶ Detectarea imaginilor folosește derivatele de ordin întâi (Gradient) și al doilea (Laplacian) care sunt implementate prin convoluție folosind filtre trece-sus.
- ▶ Analiza derivatelor:
 - ▶ Derivata de ordinul întâi (Gradient): măsoară variația intensității.
 - ▶ Derivata de ordinul al doilea (Laplacian): indică schimbarea de curbură.
- ▶ Gradientul este o metodă simplă de detecție a marginilor, bazată pe derivata de ordinul întâi.

Derivate și Margini

► Model Grafic pentru Derivate.



Derivate și Margini - Formula Gradientului

- ▶ Gradientul se poate calcula prin convoluție cu un kernel. Derivata pe axa x a unei imagini poate fi scrisă astfel:

$$\frac{\partial I(x, y)}{\partial x} = I(x, y) * K_x$$

unde K_x este kernelul pentru derivata pe axa x . Kernelul K_x este definit astfel:

$$K_x = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$$

- ▶ Similar, derivata pe axa y poate fi scrisă astfel:

$$\frac{\partial I(x, y)}{\partial y} = I(x, y) * K_y$$

unde K_y este kernelul pentru derivata pe axa y . Kernelul K_y este definit astfel:

$$K_y = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

Derivate și Margini - Detecția Marginilor

- ▶ După aplicarea operatorilor de gradient, magnitudinea gradientului se calculează astfel:

$$G = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

- ▶ Aceasta măsoară variația intensității imaginii, iar valorile mari ale magnitudinii indică margini.
- ▶ Direcția gradientului este dată de:

$$\theta = \text{atan2}\left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}\right)$$

Operatorul Sobel

- ▶ Operatorul Sobel este utilizat pentru detecția marginilor într-o imagine, fiind o variantă netezită a operatorilor Prewitt.
- ▶ Acesta utilizează două kerneluri pentru calcularea gradientului pe axele X și Y:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

- ▶ Aceste kerneluri sunt aplicate pe imagine pentru a calcula gradientul orizontal și vertical al pixelilor.
- ▶ Gradientii ∇G_x și ∇G_y sunt combinați pentru a obține mărimea gradientului:

$$\text{Magnitudine} = \sqrt{G_x^2 + G_y^2}$$

- ▶ Direcția gradientului poate fi calculată prin:

$$\theta = \text{atan2}(G_y, G_x)$$

Operatorul Canny

- ▶ Operatorul Canny este un algoritm pentru detecția marginilor care combină filtrarea Gaussiană, calculul gradientului și aplicarea unor praguri multiple pentru a identifica marginile într-o imagine.
- ▶ Formula pentru aplicarea filtrului Gaussian pe o imagine este:

$$I_G(x, y) = I(x, y) * G(\sigma)$$

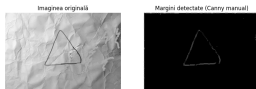
unde $I(x, y)$ reprezintă imaginea originală, iar $G(\sigma)$ este funcția de distribuție normală Gaussiană cu deviația standard σ .

- ▶ După filtrare, se calculează gradientul imaginii folosind operatorii Sobel G_x și G_y pentru a determina magnitudinea și direcția

Concluzii și Comparații: ground truth pe cod scris fără coduri predefinite

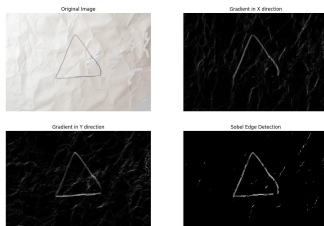
Imagine: paper4.jpeg (dimensiuni: 680 x 460)

► Rezultat Canny:



Timp de execuție: **8.516952s** secunde

► Rezultat Sobel:



Timp de execuție: **5.679149** secunde

Biblioteci Folosite

- ▶ **OpenCV**: procesare imagini eficientă.
- ▶ **Numpy**: manipulare rapidă a matricilor.
- ▶ **Matplotlib**: vizualizare și graficare rezultate.
- ▶ **Time**: măsurare eficiență algoritmi.

Concluzii

- ▶ Operatorul Canny, deși este mai costisitor din punct de vedere al resurselor (timp și energie), oferă cele mai precise rezultate în detecția imaginilor datorită eliminării zgomotului.
- ▶ Operatorul Sobel, pe de altă parte, este mai rapid și mai eficient din punct de vedere al resurselor, însă oferă rezultate mai puțin precise, fiind mai sensibil la zgomot.
- ▶ În funcție de nevoi, modul de utilizare și resurse, poate fi ales un operator diferit pentru procesarea imaginilor, întrucât fiecare are avantaje / dezavantaje

Bibliografie



Djemel Ziou and Salvatore Tabbone

Edge Detection Tehniques - An Overview.

URL: <https://inria.hal.science/inria-00098446/>



O. R. Vincent, Clausthal University of Technology, Germany
and University of Agriculture, Abeokuta, Nigeria O. Folorunso
Department of Computer Science, University of Agriculture,
Abeokuta, Nigeria

A Descriptive Algorithm for Sobel Image Edge Detection

URL: <https://proceedings.informingscience.org/InSITE2009/InSITE09p097-107Vincent613.pdf>



The University of Edinburgh

Canny Edge Detector

URL:

<https://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm>