# EDGE DETECTION: ALGORITMUL CANNY

Miu Elena Adania
Alexandrescu Tudor Alexandru

# CUPRINS

# CE ESTE EDGE DETECTION?

- Edge detection implică folosirea unei varietăți de metode matematice care au scopul de a identifica contururi în imagini.
- Contururile (edges) sunt zone în imagini care prezintă modificări bruște de contrast.
- Aceste modificări în contrast pot fi localizate pe ambele axe ale imaginii.
- Edge detection implică mai degrabă o colecție de algoritmi, și nu un algoritm de sine stătător.
- Edge detection nu este un silver bullet, parametrizarea realizându-se de la caz la caz.

# EXEMPLU DE EDGE DETECTION

Imaginea Originala

Edge Detection

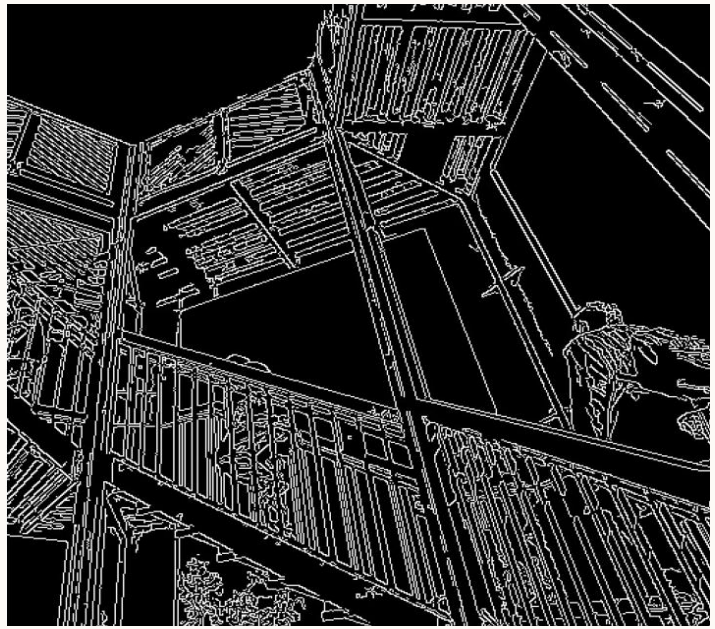

Figura 1: Edge detection utilizând algoritumul Canny

Sursa: calcule ale autorilor folosind imaginea ascent din biblioteca scipy și aplicând cv2.Canny

# JUSTIFICARE ȘI APLICAȚII PRACTICE

- Edge detection reprezintă o funcționalitate esențială în procesarea imaginilor fiind parte integrată în cadrul algoritmilor de computer vision.
- Joacă un rol deosebit în feature detection și feature extraction.
- Edge detection ne permite să obținem conturul unui obiect sau putem izola diverse obiecte (componente) de interes din imagine.
- Este utilizat în practică pentru: medical imaging, computer vision, pattern recognition, machine learning.

# ABORDAREA TEHNICĂ

Pașii Algoritmului Canny pentru Edge Detection:

1. Transformarea imaginii în formatul gray-scale
2. Reducerea zgomotului din imagine folosind filtrul Gaussian
3. Calcularea gradientului și a direcției (folosind, de exemplu, operatorul Sobel)
4. Suprimarea non-maximelor
5. Hysteresis thresholding

# REDUCEREA ZGOMOTULUI DIN IMAGINE

- Aplicăm algoritmul Gaussian smoothing (low pass filter)
- Dorim să reducem frecvențele înalte din imagini (noise)
- Ne interesează doar edge-urile și nu toate detaliile din imagine
- Aplicarea Gaussian smoothing se poate face prin 2 metode:
  - Convoluție între imagine și kernel
  - Produsul elementelor în domeniul frecvenței
- Filtrarea frecvențelor este lină (nu bruscă) deoarece ne bazăm pe forma distribuției normale și depinde de parametrul sigma
- Algoritmul pastrează edge-urile obiectelor din imagine

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Formula 1: Funcția de smoothing gaussiană

# CALCULAREA GRADIENTULUI

- Cum calculăm schimbările de intensitate la nivelul imaginii?
- Echivalent: cum calculăm variația unei funcții de 2 parametri?
- => Cu ajutorul gradientului

$$\frac{\partial I}{\partial x}(x,y) = I(x+1,y) - I(x-1,y)$$

$$\frac{\partial I}{\partial y}(x,y) = I(x,y+1) - I(x,y-1)$$

Formula 2: gradientul unui pixel

# CALCULAREA GRADIENTULUI

Calcularea Gradientului se face folosind următoarele formule:

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

Formula 3: Magnitudinea Gradientului

$$G_x = h_x * A$$

$$G_y = h_y * A$$

Formula 4: Aproximările Gradientului

,unde h este kernel-ul,

$*$ este operația de convoluție,
iar A este imaginea originală

$$\Theta = arctan\left(\frac{G_y}{G_x}\right)$$

Formula 5: Direcția Gradientului

# CALCULAREA GRADIENTULUI

Pentru kernel-ul h din formula aproximărilor gradientului putem avea mai multe tipuri de operatori:

- Operatorul Sobel
- Operatorul Prewitt
- Operatorul Roberts

De asemenea, edge detection poate folosi și calculul derivatei a doua cu Operatorul Laplacian.

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

Formula 6: hx și hy la Operatorul Sobel

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

Formula 7: hx și hy la Operatorul Prewitt

# EXEMPLU (OPERATORUL SOBEL)

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | 3 | 7 | 4 | 6 | 9 | 2 |
| 6 | 7 | 4 | 3 | 7 | 7 | 2 |
| 5 | 4 | 1 | 7 | 5 | 1 | 4 |
| 0 | 9 | 5 | 8 | 0 | 9 | 2 |
| 6 | 3 | 8 | 2 | 4 | 2 | 6 |
| 4 | 8 | 6 | 1 | 3 | 8 | 1 |
| 9 | 8 | 9 | 4 | 1 | 3 | 6 |

Imaginea originală A

$h_x$

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 2 | 0 | -2 |
| 1 | 0 | -1 |

$h_y$

| | | |
|---|---|---|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

$G_x$

| | | | | | | |
|---|---|---|---|---|---|---|
| -8 | 1 | -1 | 0 | 19 | -17 | -26 |
| -2 | -7 | -4 | 9 | 7 | -15 | -14 |
| 8 | -5 | 1 | 6 | -7 | -5 | -6 |
| 14 | 8 | 0 | -10 | -4 | 5 | -7 |
| 7 | 11 | -10 | -16 | 8 | 4 | -6 |
| 4 | 6 | -19 | -18 | 13 | 3 | -7 |
| 1 | 2 | -19 | -27 | 4 | 13 | 2 |

$G_y$

| | | | | | | |
|---|---|---|---|---|---|---|
| -4 | -5 | 3 | 4 | 1 | 3 | 2 |
| 2 | 5 | 8 | 1 | 7 | 15 | 2 |
| 16 | 1 | -9 | -4 | 7 | 3 | -2 |
| -2 | -6 | -8 | 4 | 6 | -3 | -7 |
| -11 | -3 | 6 | 10 | 2 | 0 | 4 |
| -14 | -14 | -9 | -2 | 3 | 1 | -1 |
| -15 | -8 | -9 | -7 | 6 | 7 | -10 |

# SUPRIMAREA NON-MAXIMELOR

- Echivalentul în procesarea semnalelor: high-pass filter
- Suprimarea non-maximelor reprezintă subțierea contururilor de-a lungul direcției gradientului.
- Elementele de care avem nevoie: magnitudinea și direcția gradienților.
- Verificăm pentru fiecare pixel dacă acesta reprezintă un maxim local, pe direcția gradientului.
- Dacă un pixel este maxim local, se păstrează pentru pasul următor, altfel este eliminat.

# HYSTERESIS THRESHOLDING

- Hysteresis thresholding reprezintă unirea punctelor de muchie de pe un contur (defragmentarea conturului).
- Stabilim două praguri, lowerVal și upperVal. Filtrăm pixelii cu un gradient slab (lowerVal) și păstrăm pixelii cu un gradient mare (upperVal).
- Pixelii cu o intensitate a gradientului mai mare decât upperVal fac parte sigur din contur, iar pixelii cu o valoare sub lowerVal sunt eliminați.
- Pixelii cu valoarea cuprinsă între upperVal și lowerVal sunt considerați parte din contur dacă sunt conectați direct cu un pixel care este sigur în contur. Altfel, aceștia sunt eliminați.

# TEHNOLOGII FOLOSITE

Partea de cod a proiectului va fi implementată în Python 3.11 utilizând următoarele librării:

- Numpy și Scipy pentru calcul computațional
- Matplotlib – pentru plotare
- OpenCV pentru a verifica rezultatele noastre cu implementările consacrate

# CONCLUZII

- Edge detection este un algoritm utilizat în detectarea conturului elementelor de interes dintr-o imagine.
- Este frecvent utilizat în probleme legate de feature detection și feature extraction.
- Algoritmul Canny este cel mai cunoscut algoritm de edge detection.
- Algoritmul Canny este parametrizabil în funcție de imaginea de input și scopul final.

# BIBLIOGRAFIE


- Machine Vision by E. R. Davies
- Ziou, Djemel & Tabbone, Salvatore. (1998). 'Edge detection techniques: An overview'. International Journal of Patttern Recognition and Image Analysis. 4. 537-559.
- Mutneja, Vikram. (2015). Methods of Image Edge Detection: A Review. Journal of Electrical & Electronic Systems. 04. 10.4172/2332-0796.1000150.
- OpenCV Edge Detection ( cv2.Canny )