

Procesarea Imaginilor Digitale

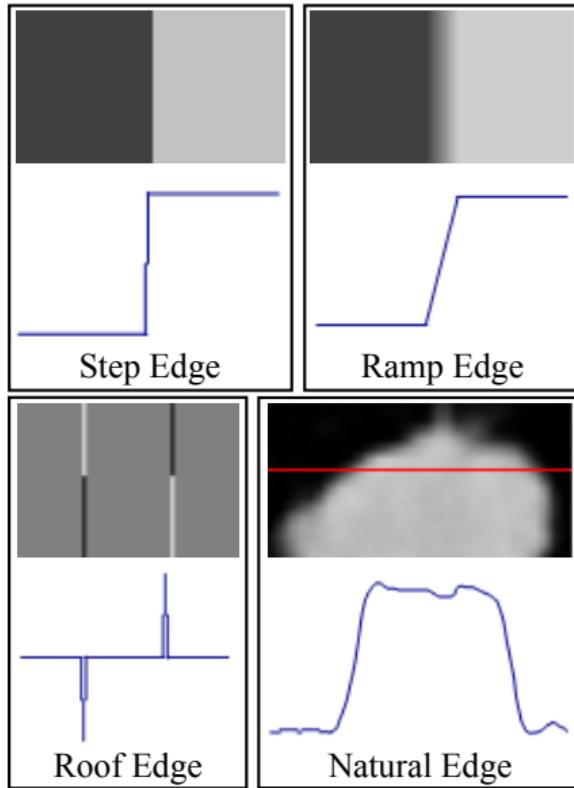
Curs - Filtre trece-sus - Detectarea de contur

Universitatea "Transilvania" din Brașov

Concepțe de bază

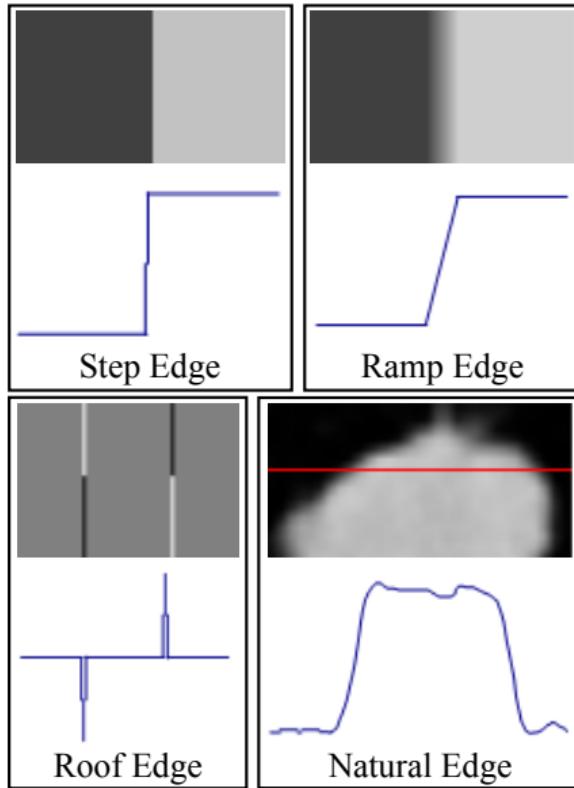
- **Margine - Edge:** - tranziție semnificativă între nivele de gri - caracter local.
- **Contur - Contour/Boundary:** - mulțime conectată de pixeli margine - concept global
- **Detectarea marginilor - Edge detection:**
 - reprezintă o etapă premergătoare pentru segmentare
 - are la bază filtrare și *thresholding*

Tipuri de margini



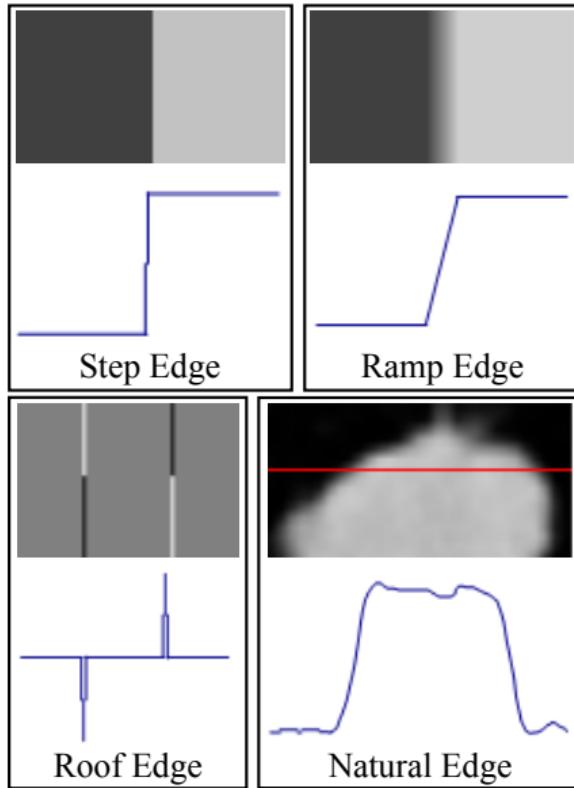
- ① **Margine treaptă - *step edge*** -
magine ideală, caracteristică
imaginilor sintetice.

Tipuri de margini



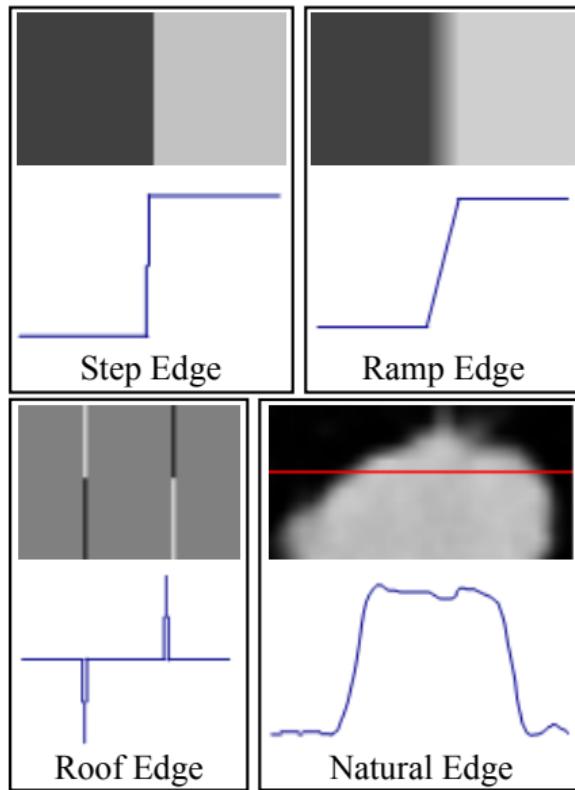
- ① **Margine treaptă - *step edge*** - margine ideală, caracteristică imaginilor sintetice.
- ② **Margine în rampă - *ramp edge*** - tranziție liniară prin tonuri de gri echidistante, apare în modele sintetice.

Tipuri de margini



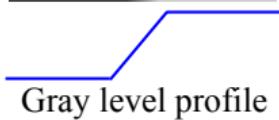
- ① **Margine treaptă - *step edge*** - margine ideală, caracteristică imaginilor sintetice.
- ② **Margine în rampă - *ramp edge*** - tranziție liniară prin tonuri de gri echidistante, apare în modele sintetice.
- ③ **Margine "acoperiș" - *roof edge*** - caracteristică liniilor dintr-o imagine.

Tipuri de margini



- ① **Margine treaptă - *step edge*** - margine ideală, caracteristică imaginilor sintetice.
- ② **Margine în rampă - *ramp edge*** - tranziție liniară prin tonuri de gri echidistante, apare în modele sintetice.
- ③ **Margine "acoperiș" - *roof edge*** - caracteristică liniilor dintr-o imagine.
- ④ **Margine realistă / naturală -** tranziția de la un ton de gri la altul nu este echidistantă.

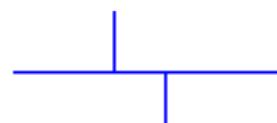
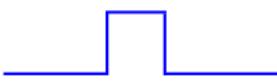
Tipuri de margini - Observații



Lățimea pentru

- margine treaptă - un pixel
- margine rampă/naturală - depinde de panta rampei.

Tipuri de margini - Observații



- **Lățimea** pentru
 - margine treaptă - un pixel
 - margine rampă/naturală - depinde de panta rampei.
- **Instrumentul matematic**: derivata de ordinul întâi / de ordinul al doilea.
- **Derivata I**: existență și orientare contur.
- **Derivata a II-a**: existența unei margini și trecerea deschis-închis și invers.

Modelarea matematică a filtrelor de tip gradient

Derivata de ordinul întâi pentru funcții bidimensionale este dată prin **gradient**:

$$\nabla f(x, y) = \text{grad}(f(x, y)) = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}$$

Aproximarea derivatelor parțiale - Notații

- **Derivata parțială în direcția 0x:**

$$\frac{\partial f(x, y)}{\partial x} \approx D_{hx}f(x, y)$$

- **Derivata parțială în direcția 0y:**

$$\frac{\partial f(x, y)}{\partial y} \approx D_{hy}f(x, y)$$

Aproximarea derivatelor parțiale

$$D_{hx} f(x, y) =$$

- Diferența înainte:

$$f_x^+(x, y) = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} = f(x + 1, y) - f(x, y)$$

Aproximarea derivatelor parțiale

$$D_{hx} f(x, y) =$$

- Diferența înainte:

$$f_x^+(x, y) = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} = f(x + 1, y) - f(x, y)$$

- Diferența înapoi:

$$f_x^-(x, y) = \frac{f(x, y) - f(x - \Delta x, y)}{\Delta x} = f(x, y) - f(x - 1, y)$$

Aproximarea derivatelor parțiale

$$D_{hx} f(x, y) =$$

- Diferența înainte:

$$f_x^+(x, y) = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} = f(x + 1, y) - f(x, y)$$

- Diferența înapoi:

$$f_x^-(x, y) = \frac{f(x, y) - f(x - \Delta x, y)}{\Delta x} = f(x, y) - f(x - 1, y)$$

- Diferența simetrică:

$$f_x^s(x, y) = \frac{f(x + \Delta x, y) - f(x - \Delta x, y)}{2\Delta x} = f(x + 1, y) - f(x - 1, y)$$

Măști de filtrare corespunzătoare derivatelor parțiale

Ce măști de filtrare corespund aproximărilor derivatei?

- **Diferența înainte:**

$$f_x^+(x, y) = f(x + 1, y) - f(x, y)$$

- **Diferența înapoi:**

$$f_x^-(x, y) = f(x, y) - f(x - 1, y)$$

- **Diferența simetrică:**

$$f_x^s(x, y) = f(x + 1, y) - f(x - 1, y)$$

centrate pe elementele marcate cu punct

Măști de filtrare corespunzătoare derivatelor parțiale

Ce măști de filtrare corespund aproximărilor derivatei?

- **Diferență înainte:**

$$f_x^+(x, y) = f(x + 1, y) - f(x, y) \quad [\begin{array}{cc} -1 & 1 \end{array}]$$

- **Diferență înapoi:**

$$f_x^-(x, y) = f(x, y) - f(x - 1, y)$$

- **Diferență simetrică:**

$$f_x^s(x, y) = f(x + 1, y) - f(x - 1, y)$$

centrate pe elementele marcate cu punct

Măști de filtrare corespunzătoare derivatelor parțiale

Ce măști de filtrare corespund aproximărilor derivatei?

- **Diferență înainte:**

$$f_x^+(x, y) = f(x + 1, y) - f(x, y) \quad [\begin{array}{cc} -1 & 1 \end{array}]$$

- **Diferență înapoi:**

$$f_x^-(x, y) = f(x, y) - f(x - 1, y) \quad [\begin{array}{cc} -1 & 1 \end{array}]$$

- **Diferență simetrică:**

$$f_x^s(x, y) = f(x + 1, y) - f(x - 1, y)$$

centrate pe elementele marcate cu punct

Măști de filtrare corespunzătoare derivatelor parțiale

Ce măști de filtrare corespund aproximărilor derivatei?

- **Diferență înainte:**

$$f_x^+(x, y) = f(x + 1, y) - f(x, y) \quad [\begin{array}{cc} -1 & 1 \end{array}]$$

- **Diferență înapoi:**

$$f_x^-(x, y) = f(x, y) - f(x - 1, y) \quad [\begin{array}{cc} -1 & 1 \end{array}]$$

- **Diferență simetrică:**

$$f_x^s(x, y) = f(x + 1, y) - f(x - 1, y) \quad [\begin{array}{ccc} -1 & 0 & 1 \end{array}]$$

centrate pe elementele marcate cu punct

Măști de filtrare corespunzătoare derivatelor parțiale

Care dintre imagini a fost filtrată cu o mască corespunzătoare derivei orizontale și care cu cea corespunzătoare derivei verticale?



Măști de filtrare corespunzătoare derivatelor parțiale



$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



$$[-1 \ 0 \ 1]$$

Observație: Derivata pe orizontală detectează margini verticale și cea pe verticală detectează margini orizontale!

Gradient

- **Gradientul:**

$$\nabla f(x, y) = \text{grad}(f(x, y)) = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}$$

- **Norma gradientului:** - "viteza" de variație a nivelului de gri

$$|\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2}$$

- **Direcția gradientului:** -pentru stabilirea direcției marginii

$$\varphi(x, y) = \arctg \left(\frac{f_y}{f_x} \right)$$

Operatori de gradient - Operatorul Sobel

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

 S_x

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

 S_y

- Operatorul S_x :

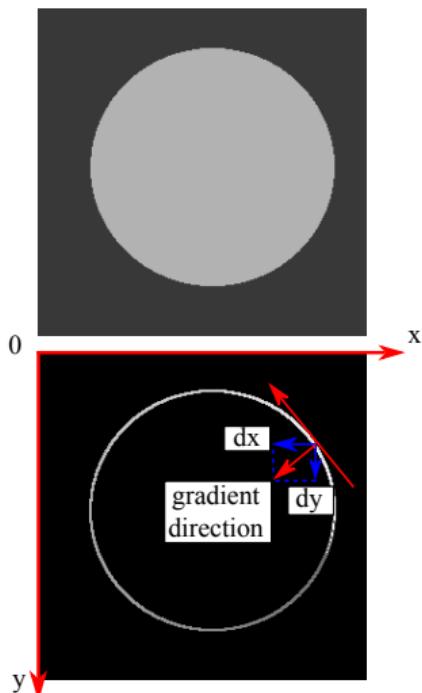
$$f_x(x, y) = f(x + 1, y - 1) - f(x - 1, y - 1) + 2 * f(x + 1, y) - 2 * f(x - 1, y) + f(x + 1, y + 1) - f(x - 1, y + 1)$$

- Operatorul S_y :

$$f_y(x, y) = f(x - 1, y + 1) - f(x - 1, y - 1) + 2 * f(x, y + 1) - 2 * f(x, y - 1) + f(x + 1, y + 1) - f(x + 1, y - 1)$$

- Norma gradientului: $\sqrt{f_x^2 + f_y^2}$

Sobel direcțional



- Direcția gradientului este perpendiculară pe direcția conturului.
- Pot fi detectate contururi după o anumită direcție.
- Pentru calcul - utilizarea funcției $\text{atan2}(f_y, f_x)$ sau $\text{atan}(f_y/f_x)$

Aplicarea operatorului Sobel

Pași

- 1 Calculare $f_x(x, y) = (S_x * f)(x, y)$ și $f_y(x, y) = (S_y * f)(x, y)$.

Aplicarea operatorului Sobel

Pași

- 1 Calculare $f_x(x, y) = (S_x * f)(x, y)$ și $f_y(x, y) = (S_y * f)(x, y)$.
- 2 Calculare imagine gradient $Grad(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)}$.

Aplicarea operatorului Sobel

Pași

- ① Calculare $f_x(x, y) = (S_x * f)(x, y)$ și $f_y(x, y) = (S_y * f)(x, y)$.
- ② Calculare imagine gradient $Grad(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)}$.
- ③ Calculare - dacă este necesar - a direcției marginilor.

Aplicarea operatorului Sobel

Pași

- ① Calculare $f_x(x, y) = (S_x * f)(x, y)$ și $f_y(x, y) = (S_y * f)(x, y)$.
- ② Calculare imagine gradient $Grad(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)}$.
- ③ Calculare - dacă este necesar - a direcției marginilor.
- ④ *Thresholding* pentru obținerea contururilor.

Operatorul Sobel - Exemplu

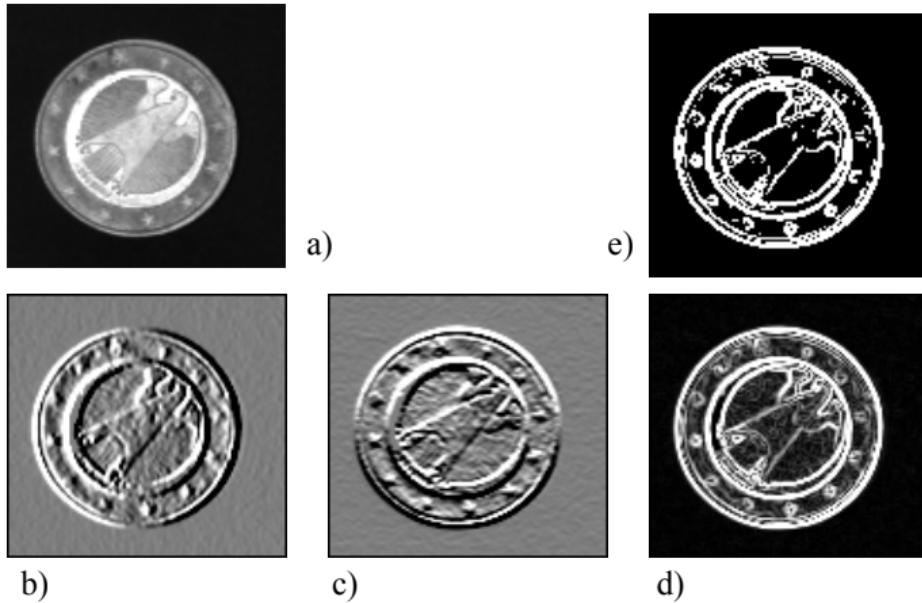


Figure: a) Imagine originală; b) Imagine după aplicarea operatorului S_x ; c) Imagine după aplicarea operatorului S_y ; e) Imagine gradient; f) Imagine binară după aplicarea unui prag.

Operatorul Canny - Obiective

- Reducerea ratei de eroare în detectarea pixelilor de contur.
- Localizarea corectă a conturului detectat, cât mai aproape de conturul real.
- Obținerea unor contururi de grosime un pixel.

Operatorul Canny - Etape

- ① ***Smoothing*** - Reducere zgromadire - filtrare gaussiană

Operatorul Canny - Etape

- ① ***Smoothing*** - Reducere zgomot - filtrare gaussiană
- ② ***High-pass filtering*** - Calcularea imaginii gradient cu operatorul Sobel.

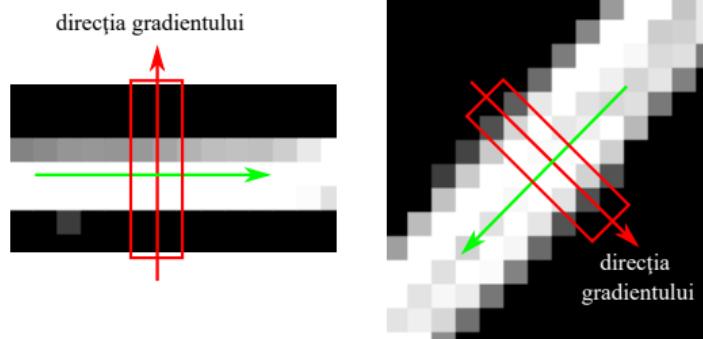
Operatorul Canny - Etape

- ① ***Smoothing*** - Reducere zgomot - filtrare gaussiană
- ② ***High-pass filtering*** - Calcularea imaginii gradient cu operatorul Sobel.
- ③ ***Nonmaxima Suppresion*** - Subtire contururi la grosime 1

Operatorul Canny - Etape

- ① ***Smoothing*** - Reducere zgomot - filtrare gaussiană
- ② ***High-pass filtering*** - Calcularea imaginii gradient cu operatorul Sobel.
- ③ ***Nonmaxima Supresion*** - Subtire contururi la grosime 1
- ④ ***Hysteresys thresholding*** - Thresholding cu două praguri și conectarea contururilor slabe.

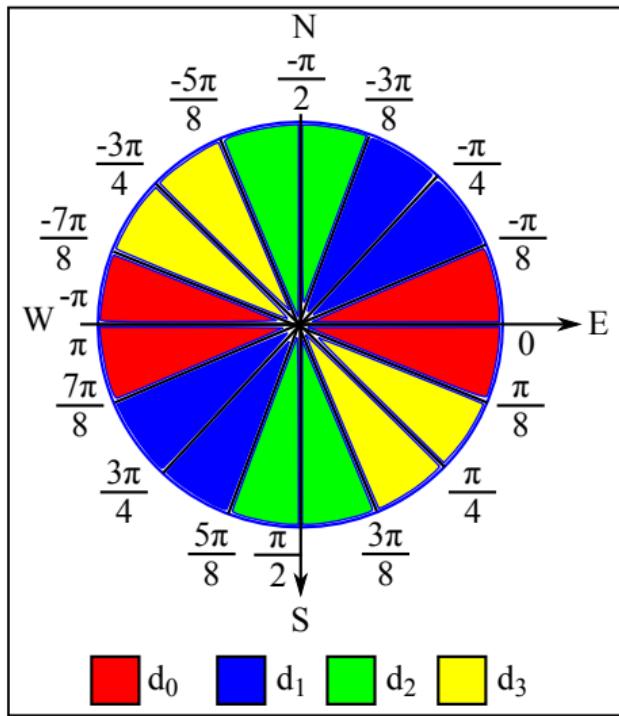
Operatorul Canny - Nonmaxima Supresion



Problema: Sobel produce contururi "groase". Vrem subțierea:

- trebuie subțiat în direcția perpendiculară pe contur
- gradientul este perpendicular pe contur.
⇒ trebuie subțiat în direcția gradientului ⇒ pentru fiecare pixel de contur trebuie calculată direcția gradientului.

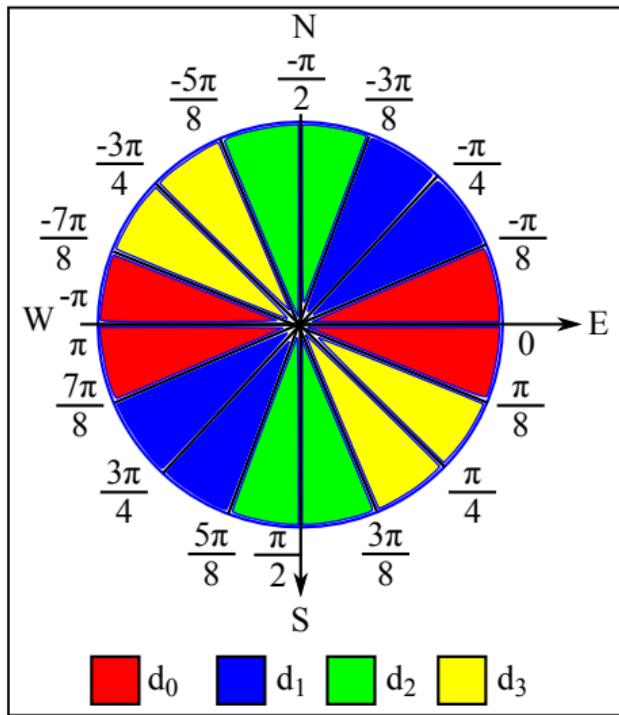
Operatorul Canny - Nonmaxima Supresion



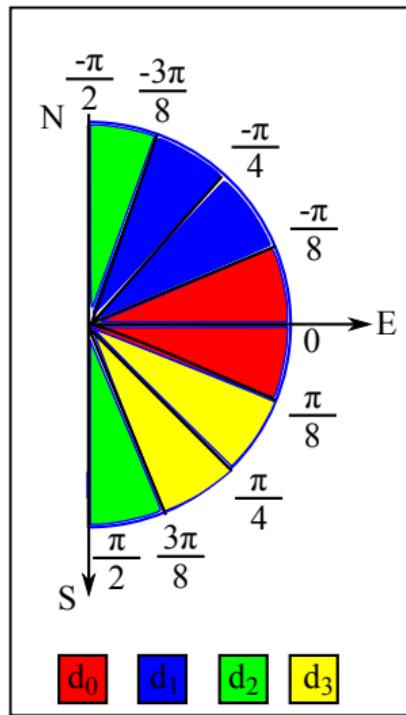
Direcția gradientului: dată prin unghiul α față de Ox , $\alpha \in [-\pi, \pi]$.

Observație: prea multe direcții!

Operatorul Canny - Nonmaxima Supresion

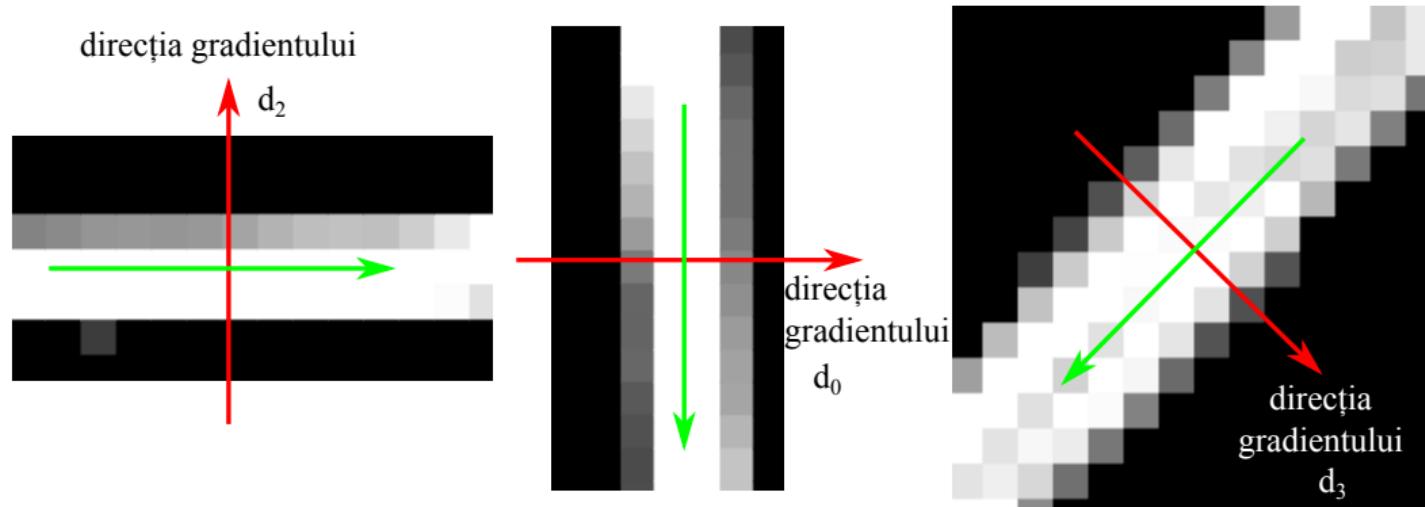


Operatorul Canny - *Nonmaxima Supresion*



Utilizarea funcției $\text{atan}(f_y/f_x)$ care produce rezultate în intervalul $[-\pi/2, \pi/2]$.

Operatorul Canny - Nonmaxima Supresion



Operatorul Canny - *Hysteresys Tresholding*

- ① Se stabilesc două praguri $T_1 < T_2$

Operatorul Canny - *Hysteresys Tresholding*

- ① Se stabilesc două praguri $T_1 < T_2$
- ② Pentru $|Grad(x, y)| \leq T_1 \Rightarrow g(x, y) = 0$

Operatorul Canny - *Hysteresys Tresholding*

- ① Se stabilesc două praguri $T_1 < T_2$
- ② Pentru $|Grad(x, y)| \leq T_1 \Rightarrow g(x, y) = 0$
- ③ Pentru $|Grad(x, y)| > T_2 \Rightarrow g(x, y) = 255$

Operatorul Canny - *Hysteresys Tresholding*

- ① Se stabilesc două praguri $T_1 < T_2$
- ② Pentru $|Grad(x, y)| \leq T_1 \Rightarrow g(x, y) = 0$
- ③ Pentru $|Grad(x, y)| > T_2 \Rightarrow g(x, y) = 255$
- ④ Pentru $T_1 < |Grad(x, y)| \leq T_2$

Operatorul Canny - *Hysteresys Tresholding*

- ① Se stabilesc două praguri $T_1 < T_2$
- ② Pentru $|Grad(x, y)| \leq T_1 \Rightarrow g(x, y) = 0$
- ③ Pentru $|Grad(x, y)| > T_2 \Rightarrow g(x, y) = 255$
- ④ Pentru $T_1 < |Grad(x, y)| \leq T_2$
 - dacă pixelul (x, y) conectat cu un pixel (s, t) cu gradientul $> T_2 \Rightarrow g(x, y) = 255$

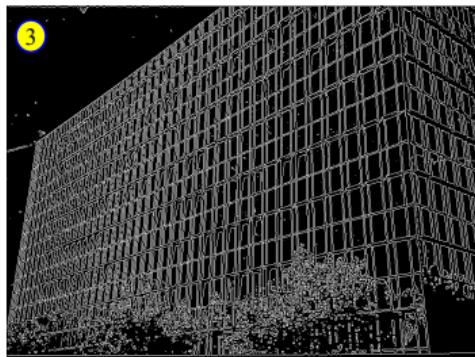
Operatorul Canny - *Hysteresys Tresholding*

- ① Se stabilesc două praguri $T_1 < T_2$
- ② Pentru $|Grad(x, y)| \leq T_1 \Rightarrow g(x, y) = 0$
- ③ Pentru $|Grad(x, y)| > T_2 \Rightarrow g(x, y) = 255$
- ④ Pentru $T_1 < |Grad(x, y)| \leq T_2$
 - dacă pixelul (x, y) conectat cu un pixel (s, t) cu gradientul $> T_2 \Rightarrow g(x, y) = 255$
 - altfel $g(x, y) = 0$

Operatorul Canny - Exemplu



1. Imagine originală
2. Sobel $T=100$
3. Canny $T_1=40$, $T_2=60$



Deivata de ordinul II - Operatorul Laplace

Derivata de ordinul II în direcția 0x:

$$\frac{\partial^2 f}{\partial x^2}(x, y) = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x}(x, y) \right) \text{ dif } \stackrel{\text{inainte}}{=}$$

Derivata de ordinul II - Operatorul Laplace

Derivata de ordinul II în direcția $0x$:

$$\frac{\partial^2 f}{\partial x^2}(x, y) = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x}(x, y) \right) \text{ dif înainte} =$$

$$= \frac{\partial}{\partial x} (f(x + 1, y) - f(x, y)) \text{ dif înapoi} =$$

Deivata de ordinul II - Operatorul Laplace

Derivata de ordinul II în direcția 0x:

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2}(x, y) &= \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x}(x, y) \right) \text{ dif } \overset{\text{inainte}}{=} \\ &= \frac{\partial}{\partial x} (f(x+1, y) - f(x, y)) \text{ dif } \overset{\text{înapoi}}{=} \\ &= (f(x+1, y) - f(x, y) - f(x, y) + f(x-1, y)) = \\ &= f(x+1, y) + f(x-1, y) - 2f(x, y)\end{aligned}$$

Derivata de ordinul II - Operatorul Laplace

Derivata de ordinul II în direcția $0x$:

$$\frac{\partial^2 f}{\partial x^2}(x, y) = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x}(x, y) \right) \text{ dif înainte} =$$

$$= \frac{\partial}{\partial x} (f(x + 1, y) - f(x, y)) \text{ dif înapoi} =$$

$$= (f(x + 1, y) - f(x, y) - f(x, y) + f(x - 1, y)) = \\ = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

Analog derivata de ordinul II în direcția $0y$:

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

Deivata de ordinul II - Operatorul Laplace

Operatorul Laplace:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2}(x, y) + \frac{\partial^2 f}{\partial y^2}(x, y) =$$

Deivata de ordinul II - Operatorul Laplace

Operatorul Laplace:

$$\begin{aligned}\nabla^2 f(x, y) &= \frac{\partial^2 f}{\partial x^2}(x, y) + \frac{\partial^2 f}{\partial y^2}(x, y) = \\ &= f(x+1, y) + f(x-1, y) - 2f(x, y) + f(x, y+1) + f(x, y-1) - 2f(x, y) =\end{aligned}$$

Deivata de ordinul II - Operatorul Laplace

Operatorul Laplace:

$$\begin{aligned}\nabla^2 f(x, y) &= \frac{\partial^2 f}{\partial x^2}(x, y) + \frac{\partial^2 f}{\partial y^2}(x, y) = \\&= f(x+1, y) + f(x-1, y) - 2f(x, y) + f(x, y+1) + f(x, y-1) - 2f(x, y) = \\&= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)\end{aligned}$$

Operatorul Laplace - Măști de filtrare

0	1	0
1	-4	1
0	1	0

a)

1	1	1
1	-8	1
1	1	1

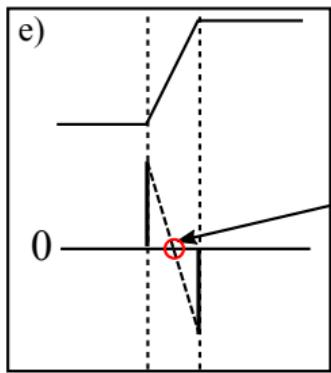
b)

0	-1	0
-1	4	-1
0	-1	0

c)

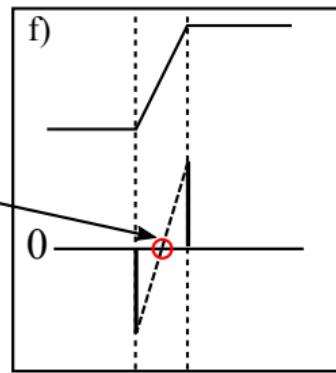
-1	-1	-1
-1	8	-1
-1	-1	-1

d)



Contur rampă

zero crossing
Laplacian



Operatorul Laplace - Exemplu



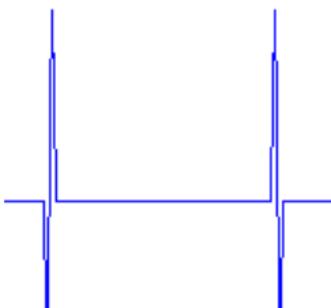
a)



b)



d)



c)

Exemplu utilizând o mască Laplace cu valoarea centrală pozitivă.

Zero-Crossings - Algoritm de detectare

A ₁	A ₂	A ₃
	(x,y)	
B ₁	B ₂	B ₃

B ₁	B ₂	B ₃
	(x,y)	
A ₁	A ₂	A ₃

A ₁		B ₁
A ₂	(x,y)	B ₂
A ₃		B ₃

B ₁		A ₁
B ₂	(x,y)	A ₂
B ₃		A ₃

A ₂	A ₃	
A ₁	(x,y)	B ₃
	B ₁	B ₂

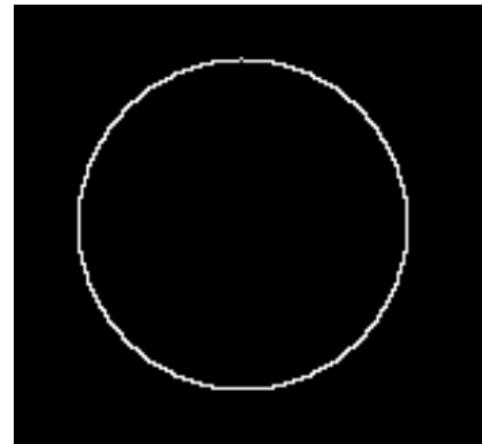
B ₂	B ₃	
B ₁	(x,y)	A ₃
	A ₁	A ₂

	A ₁	A ₂
B ₁	(x,y)	A ₃
B ₂	B ₃	

	B ₁	B ₂
A ₁	(x,y)	B ₃
A ₂	A ₃	

Dacă $\exists i, 1 \leq i \leq 3$ pentru care ($A_i \geq T$ și $B_i \leq -T$) într-una din măști $\Rightarrow (x, y)$ este zero-crossing, $T = \text{prag}$.

Zero-Crossings - Exemplu



Operatorul Marr-Hildreth - LoG

Combină filtrarea tece-jos Gaussiană cu operatorul Laplace:

$$\nabla^2 G(x, y) = \frac{\partial^2 G}{\partial x^2}(x, y) + \frac{\partial^2 G}{\partial y^2}(x, y)$$

G este funcția Gaussiană:

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

În urma derivării se obține operatorul LoG:

$$\nabla^2 G(x, y) = \frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Pe baza acestei formule se poate calcula o mască de filtrare.

Operatorul Marr-Hildreth - LoG

- ① Netezire cu un filtru trece-jos Gaussian
- ② Aplicarea operatorului Laplace
- ③ Detectare *zero-crossings*

Pașii 1 și 2 pot fi efectuați într-unul singur prin precalcularea măştii LoG

Operatorul Marr-Hildreth - Exemplu

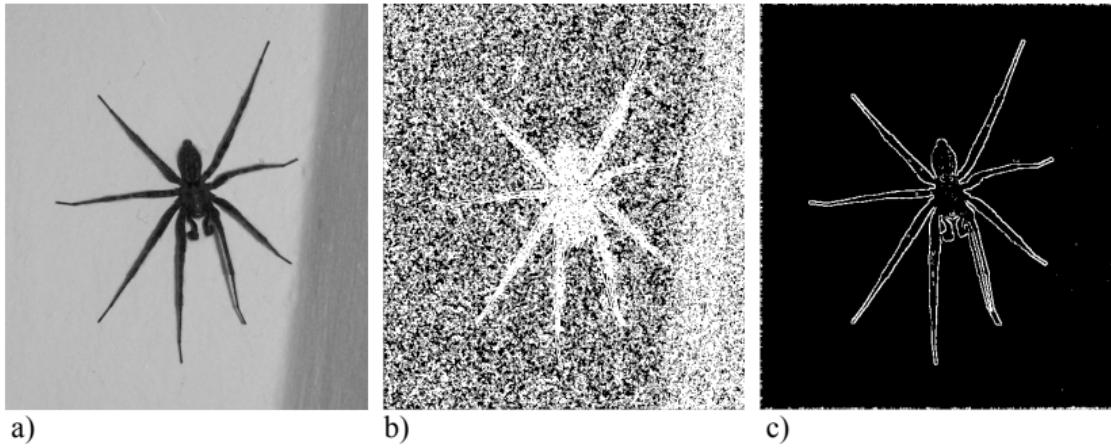


Figure: a) Imagine inițială; b) *Zero-crossings* fără netezire prealabilă; c) *Zero-crossings* după LoG

Operatorul Laplace - *Unsharp Masking*

Formulă generală

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{dacă centrul măştii e negativ} \\ f(x, y) + \nabla^2 f(x, y) & \text{dacă centrul măştii e pozitiv} \end{cases}$$

Poate fi realizat prin filtrarea cu o mască (cazul cu centrul măştii pozitiv):

Operatorul Laplace - *Unsharp Masking*

Formulă generală

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{dacă centrul măştii e negativ} \\ f(x, y) + \nabla^2 f(x, y) & \text{dacă centrul măştii e pozitiv} \end{cases}$$

Poate fi realizat prin filtrarea cu o mască (cazul cu centrul măştii pozitiv):

$$\begin{vmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{vmatrix}$$

Variante: prin creșterea valorii din centru (de ex. în loc de 5, 7 sau 9) se poate accentua efectul.

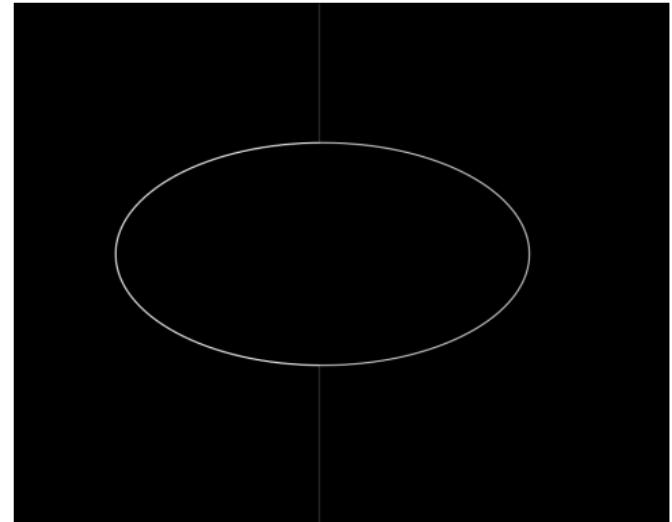
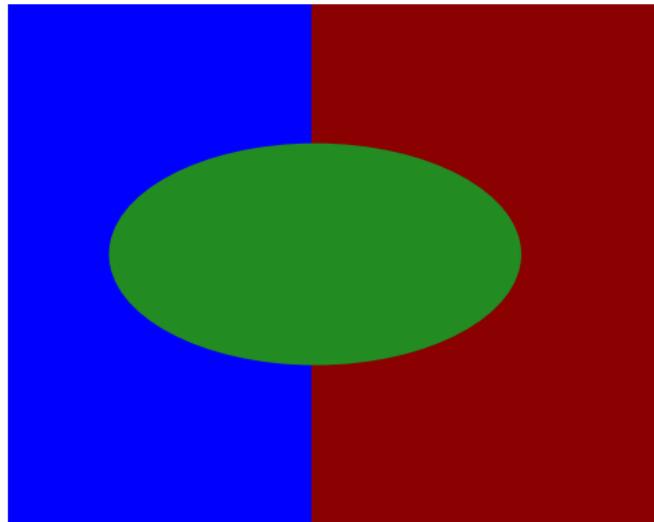
Unsharp Masking cu operatorul Laplace - Exemplu



Figure: Stânga: Imagine inițială; Dreapta: Imaginea după aplicarea operatorului de *[Unsharp Mask]*

Filtrarea trece sus a imaginilor color

Varinata 1: imaginea se face grayscale si se aplica Sobel pe grayscale.



Problema: o astfel de tehnică nu eușește să detecteze bine margini între suprafete cu aceeași intensitate luminoasă.

Filtrarea trece sus a imaginilor color - 3 canale

Varianta 2: Considerând cele 3 canale de culoare separat. Se calculează pentru fiecare canal de culoare gradientul, pe baza operatorului Sobel

$$\nabla R(x, y) = \left(\frac{\partial R(x, y)}{\partial x}, \frac{\partial R(x, y)}{\partial y} \right)^T$$

$$\nabla G(x, y) = \left(\frac{\partial G(x, y)}{\partial x}, \frac{\partial G(x, y)}{\partial y} \right)^T$$

$$\nabla B(x, y) = \left(\frac{\partial B(x, y)}{\partial x}, \frac{\partial B(x, y)}{\partial y} \right)^T$$

Filtrarea trece sus a imaginilor color - 3 canale

Varianta 2: Considerând cele 3 canale de culoare separat. Se calculează pentru fiecare canal de culoare gradientul, pe baza operatorului Sobel

$$\nabla R(x, y) = \left(\frac{\partial R(x, y)}{\partial x}, \frac{\partial R(x, y)}{\partial x} \right)^T$$

$$\nabla G(x, y) = \left(\frac{\partial G(x, y)}{\partial x}, \frac{\partial G(x, y)}{\partial x} \right)^T$$

$$\nabla B(x, y) = \left(\frac{\partial B(x, y)}{\partial x}, \frac{\partial B(x, y)}{\partial x} \right)^T$$

Se estimează pe fiecare canal de culoare cât de semnificativă este marginea pe canalul respectiv considerând norma gradientului $E_k(x, y) = \|\nabla k(x, y)\|_2$, $k \in \{R, G, B\}$, unde $\|\cdot\|_2$ reprezintă norma L_2 , dată prin radical din suma pătratelor componentelor vectorului (notăm E de la edge).

Filtrarea trece sus a imaginilor color - 3 canale

Având vectorul cu normele gradienților pe fiecare canal de culoare:

$$\text{Edge}(x, y) = \begin{pmatrix} E_R(x, y) \\ E_G(x, y) \\ E_B(x, y) \end{pmatrix}$$

- ① putem considera în imaginea rezultat culoarea dată de $\text{Edge}(x, y)$ - acesta este rezultatul oferit de exemplu de Photoshop

Filtrarea trece sus a imaginilor color - 3 canale

Având vectorul cu normele gradienților pe fiecare canal de culoare:

$$\text{Edge}(x, y) = \begin{pmatrix} E_R(x, y) \\ E_G(x, y) \\ E_B(x, y) \end{pmatrix}$$

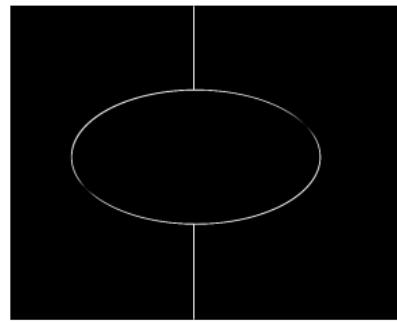
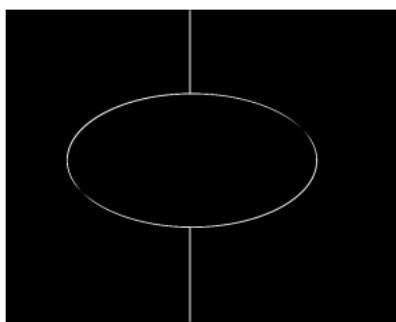
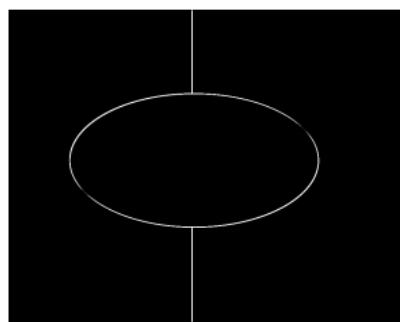
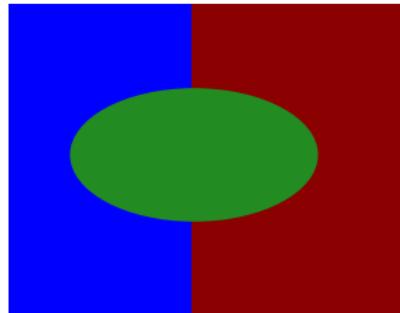
- ① putem considera în imaginea rezultat culoarea dată de $\text{Edge}(x, y)$ - acesta este rezultatul oferit de exemplu de Photoshop
- ② putem calcula o *tărie combinată* a conturului prin:

$$(2.1) \quad E_2(x, y) = \|\text{Edge}(x, y)\|_2 = \sqrt{E_R(x, y)^2 + E_G(x, y)^2 + E_B(x, y)^2}$$

$$(2.2) \quad E_1(x, y) = \|\text{Edge}(x, y)\|_1 = |E_R(x, y)| + |E_G(x, y)| + |E_B(x, y)|$$

$$(2.3) \quad E_\infty(x, y) = \|\text{Edge}(x, y)\|_\infty = \max(|E_R(x, y)|, |E_G(x, y)|, |E_B(x, y)|)$$

Filtrarea trece sus a imaginilor color - 3 canale - Exemplul 1



Pentru detectarea marginilor în imaginea color sintetică s-a aplicat metoda gradientului combinat prin norma L_1 (stânga), norma L_2 (centru) și norma L_∞ (dreapta).

Filtrarea trece sus a imaginilor color - comparație - Exemplul 2



Pentru detectarea marginilor în imaginea color sintetică s-a aplicat metoda transformarea în grayscale (dreapta sus) și gradientului combinat prin norma L_1 (stânga jos), norma L_2 (centru jos) și norma L_∞ (dreapta jos).

Gradientul în imagini RGB

Problema: la metoda prezentată mai sus, o problemă este stabilirea direcției gradientului.

O variantă: considerarea direcției gradientului pentru canalul de culoare, care are cea mai mare normă a gradientului. (aproximare)

Varianta 2: - gradient multidimensional → Jacobian

Gradientul în imagini RGB

Derivatele parțiale:

$$f_x(x, y) = \frac{\partial f(x, y)}{\partial x} = \begin{pmatrix} \frac{\partial R(x, y)}{\partial x} \\ \frac{\partial G(x, y)}{\partial x} \\ \frac{\partial B(x, y)}{\partial x} \end{pmatrix}$$

$$f_y(x, y) = \frac{\partial f(x, y)}{\partial y} = \begin{pmatrix} \frac{\partial R(x, y)}{\partial y} \\ \frac{\partial G(x, y)}{\partial y} \\ \frac{\partial B(x, y)}{\partial y} \end{pmatrix}$$

Gradientul în imagini RGB

Jacobian Matix:

$$J_f(x, y) = \begin{pmatrix} f_x(x, y) & f_y(x, y) \end{pmatrix} = \begin{pmatrix} \frac{\partial R(x, y)}{\partial x} & \frac{\partial R(x, y)}{\partial y} \\ \frac{\partial G(x, y)}{\partial x} & \frac{\partial G(x, y)}{\partial y} \\ \frac{\partial B(x, y)}{\partial x} & \frac{\partial B(x, y)}{\partial y} \end{pmatrix}$$

- $\|f_x(x, y)\|$ - variația în direcția orizontală în (x, y)
- $\|f_y(x, y)\|$ - variația în direcția verticală în (x, y)

Direcția de variație maximă

Direcția de variație maximă în punctul (x, y) = unghiul θ pentru care *gradientul direcțional* este maxim, adică acel θ pentru care este maxim:

$$grad_{\theta}(x, y) = J_f(x, y) \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} = f_x(x, y) \cos \theta + f_y(x, y) \sin \theta$$

Determinarea direcției de variație maximă

Se consideră matricea

$$M = J_f^T * J_f = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}$$

unde

$$f_{xx}(x, y) = f_x^T f_x = \left| \frac{\partial R(x, y)}{\partial x} \right|^2 + \left| \frac{\partial G(x, y)}{\partial x} \right|^2 + \left| \frac{\partial B(x, y)}{\partial x} \right|^2$$

$$f_{yy}(x, y) = f_y^T f_y = \left| \frac{\partial R(x, y)}{\partial y} \right|^2 + \left| \frac{\partial G(x, y)}{\partial y} \right|^2 + \left| \frac{\partial B(x, y)}{\partial y} \right|^2$$

$$f_{xy}(x, y) = f_x^T f_y = \frac{\partial R(x, y)}{\partial x} \frac{\partial R(x, y)}{\partial y} + \frac{\partial G(x, y)}{\partial x} \frac{\partial G(x, y)}{\partial y} + \frac{\partial B(x, y)}{\partial x} \frac{\partial B(x, y)}{\partial y}$$

Determinarea direcției de variație maximă

Direcția de variație maximă a gradientului: - vectorul propriu asociat celei mai mari valori proprii a matricii M

Variația maximă: - echivalentul normei gradientului - valoarea proprie maximă a matricii M .

$$M = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}$$

Calculul vectorilor și valorilor proprii

Calcul:

- valori proprii: - soluțiile λ_1, λ_2 ale ecuației

$$|(M - \lambda I)| = 0,$$

I - matricea identică.

- vectorii proprii corespunzători V_1, V_2 :

$$MV_1 = \lambda_1 V_1$$

$$MV_2 = \lambda_2 V_2$$

Direcția de variație maximă

Valorile proprii ale matricii :

$$\lambda_{1,2} = \frac{1}{2} \left(f_{xx} + f_{yy} \pm \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \right)$$

Evident, valoarea proprie maximă:

$$\lambda_1 = \frac{1}{2} \left(f_{xx} + f_{yy} + \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \right)$$

Direcția de variație maximă

Valoarea proprie maximă:

$$\lambda_1 = \frac{1}{2} \left(f_{xx} + f_{yy} + \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \right)$$

Vector propriu asociat valorii λ_1 :

$$v_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} f_{xx} - f_{yy} + \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \\ 2f_{xy} \end{pmatrix}$$

Orientarea vectorului v_1 :

$$\theta = \text{atan}\left(\frac{y_1}{x_1}\right) = \frac{1}{2} \text{atan}\left(\frac{2f_{xy}}{f_{xx} - f_{yy}}\right)$$

Imaginea de gradient - practic

Etape:

- Se calculează derivatele parțiale utilizând de exemplu operatorului Sobel aplicat pe fiecare componentă de culoare.

Imaginea de gradient - practic

Etape:

- Se calculează derivatele parțiale utilizând de exemplu operatorului Sobel aplicat pe fiecare componentă de culoare.
- Pentru fiecare pixel (x, y) se calculează λ_1 conform formulei

Imaginea de gradient - practic

Etape:

- Se calculează derivatele parțiale utilizând de exemplu operatorului Sobel aplicat pe fiecare componentă de culoare.
- Pentru fiecare pixel (x, y) se calculează λ_1 conform formulei
- Variația maximă este dată de $grad(x, y) = \sqrt{\lambda_1}$

Imaginea de gradient - practic

Etape:

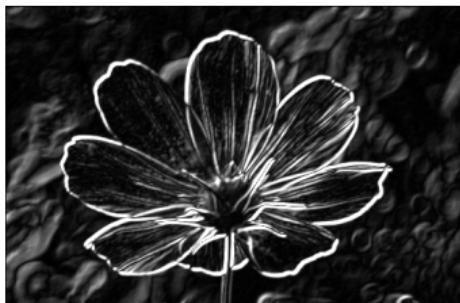
- Se calculează derivatele parțiale utilizând de exemplu operatorului Sobel aplicat pe fiecare componentă de culoare.
- Pentru fiecare pixel (x, y) se calculează λ_1 conform formulei
- Variația maximă este dată de $grad(x, y) = \sqrt{\lambda_1}$
- Dacă este necesar (ex. Canny) se calculează direcția de variație maximă (formulă)

Imaginea de gradient - practic

Etape:

- Se calculează derivatele parțiale utilizând de exemplu operatorului Sobel aplicat pe fiecare componentă de culoare.
- Pentru fiecare pixel (x, y) se calculează λ_1 conform formulei
- Variația maximă este dată de $grad(x, y) = \sqrt{\lambda_1}$
- Dacă este necesar (ex. Canny) se calculează direcția de variație maximă (formulă)
- Dacă $grad(x, y) > T$, T - threshold, atunci se consideră (x, y) pixel de contur, altfel se consideră pixel de fundal.

Gradient - Exemplu



Canny - imagini RGB

Canny - Etape:

- Reducerea zgomotului - filtru Gauss pe fiecare canal ($\sigma = 1$ sau 1.5 suficient)

Canny - imagini RGB

Canny - Etape:

- Reducerea zgomotului - filtru Gauss pe fiecare canal ($\sigma = 1$ sau 1.5 suficient)
- Se calculează matricea de direcții de variație maximă $\theta(x, y)$ + reducerea la 4 direcții principale

Canny - imagini RGB

Canny - Etape:

- Reducerea zgomotului - filtru Gauss pe fiecare canal ($\sigma = 1$ sau 1.5 suficient)
- Se calculează matricea de direcții de variație maximă $\theta(x, y)$ + reducerea la 4 direcții principale
- Se calculează imaginea gradient (norma gradientului) - cu valorile proprii corespunzătoare matricei de gradient (vezi formulă)

Canny - imagini RGB

Canny - Etape:

- Reducerea zgomotului - filtru Gauss pe fiecare canal ($\sigma = 1$ sau 1.5 suficient)
- Se calculează matricea de direcții de variație maximă $\theta(x, y)$ + reducerea la 4 direcții principale
- Se calculează imaginea gradient (norma gradientului) - cu valorile proprii corespunzătoare matricei de gradient (vezi formulă)
- Se aplică pasul de *Non-maxima suppression* - ca la algoritmul pe *grayscale*

Canny - imagini RGB

Canny - Etape:

- Reducerea zgomotului - filtru Gauss pe fiecare canal ($\sigma = 1$ sau 1.5 suficient)
- Se calculează matricea de direcții de variație maximă $\theta(x, y)$ + reducerea la 4 direcții principale
- Se calculează imaginea gradient (norma gradientului) - cu valorile proprii corespunzătoare matricei de gradient (vezi formulă)
- Se aplică pasul de *Non-maxima suppression* - ca la algoritmul pe *grayscale*
- Se aplică pasul de *Hysteresis thresholding* - ca la algoritmul pe *grayscale*

Canny - Exemplu

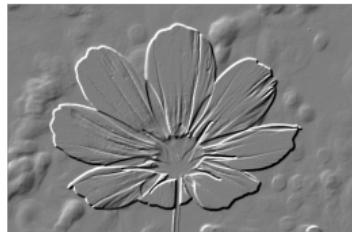


Alte filtre

Emboss Se aplică filtrul + se adaugă: 128

-1	-1	0
-1	0	1
0	1	1

-1	-1	-1	-1	0
-1	-1	-1	0	1
-1	-1	0	1	1
-1	0	1	1	1
0	1	1	1	1



Rezumat

- Margine -*Edge* - tipuri de margini
- Operatori bazați pe gradient
 - Măști de filtrare bazate pe gradient: Prewitt, Sobel, Roberts
 - Operatorul Canny
- Operatorul Laplace
 - Măști de filtrare
 - Operatorul Marr-Hildreth
 - *Unsharp Mask*
- Detectarea de contur pentru imagini color
 - Culori asemănătoare
 - Sobel pe color
 - Direcția de variație maximă
 - Canny