



Technical University of Cluj - Napoca
Computer Science Department

Procesarea Imaginilor

Curs 10

Detecția muchiilor



La ce ne folosește detecția de muchii ?

- Se pare că sistemul vizual al omului folosește muchiile ca elemente primare în procesul de recunoaștere, după care se inferează informații complexe legate de culoare/textură/formă
 - Este posibil să recunoaștem forme/obiecte doar pe baza conturului (ex. desene alb/negru, benzi desenate alb/negru în care este prezentă doar informația de contur).
- ⇒ Detecția muchiilor este o etapă care poate fi folosită în analiza automata a imaginilor
- ⇒ Detecția muchiilor - proces de segmentare

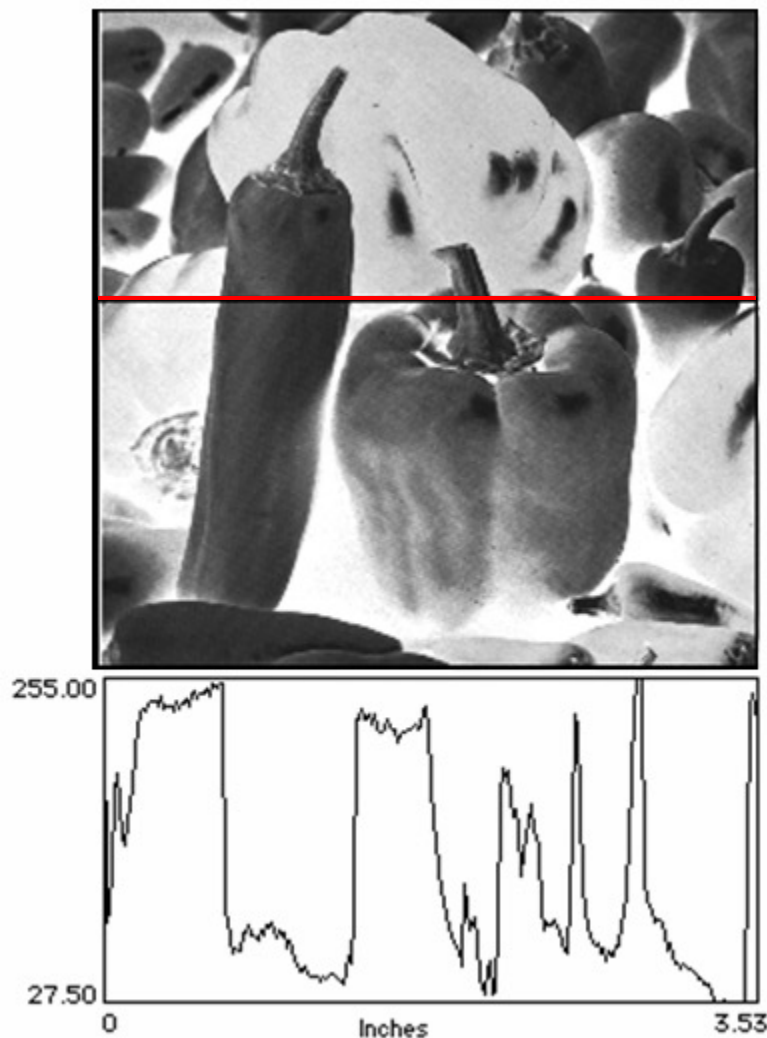
Segmentare : din informația de nivel scăzut a imaginii (pixeli / raw data) se extrage informația de nivel ridicat:

- puncte de muchie ⇒ contur ⇒ trasături de formă ⇒ analiză
- puncte de muchie ⇒ trăsături pt. stereoreconstrucție (indicator al relevanței punctelor)

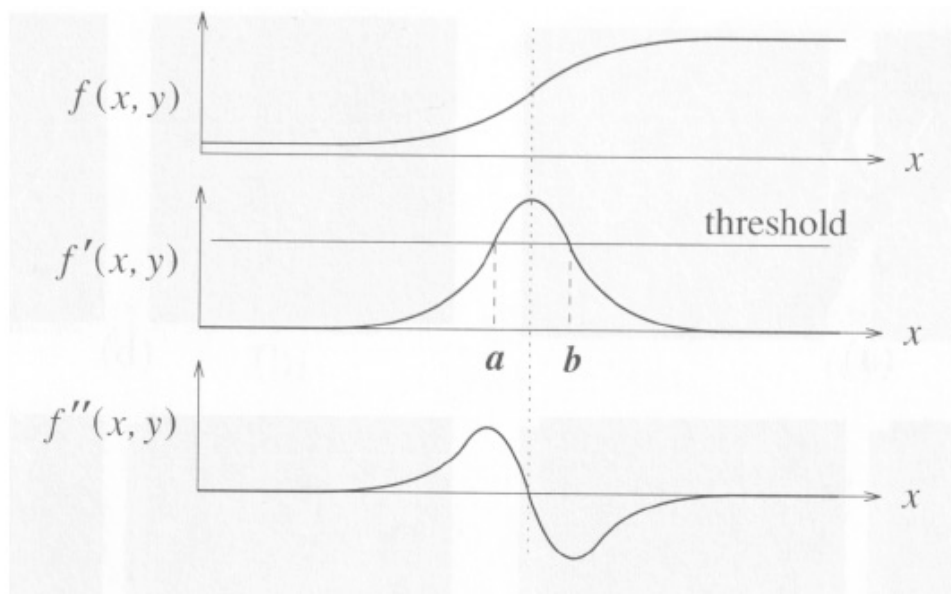


Definiție

Muchie := frontiera care separă două regiuni de intensitate diferită (intensitatea variază brusc între cele două regiuni).

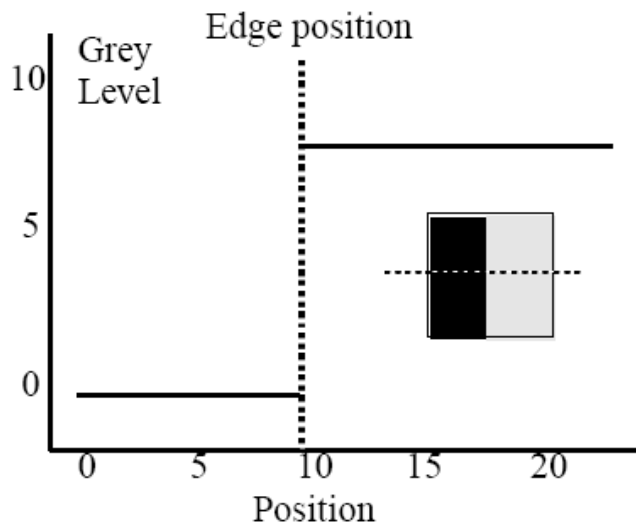


Cum putem detecta o muchie?

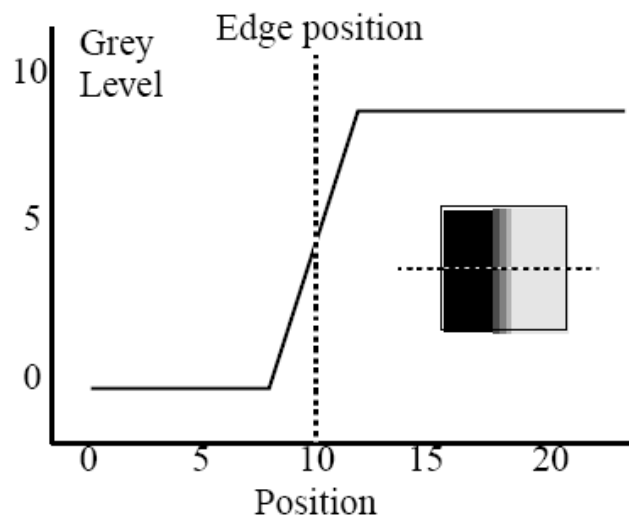




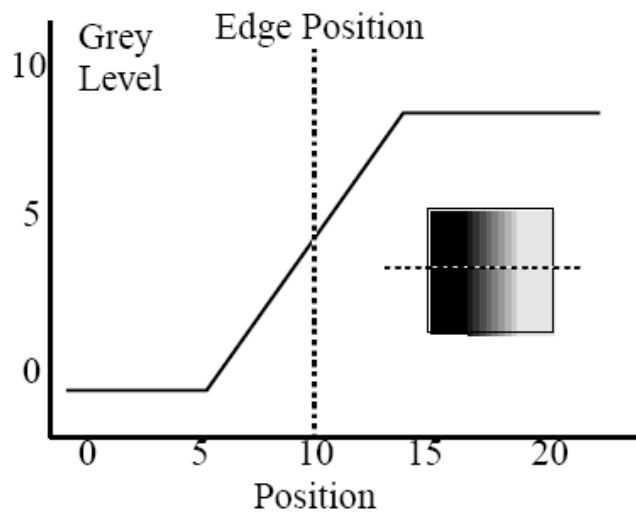
Profilul intensității în punctele de muchie



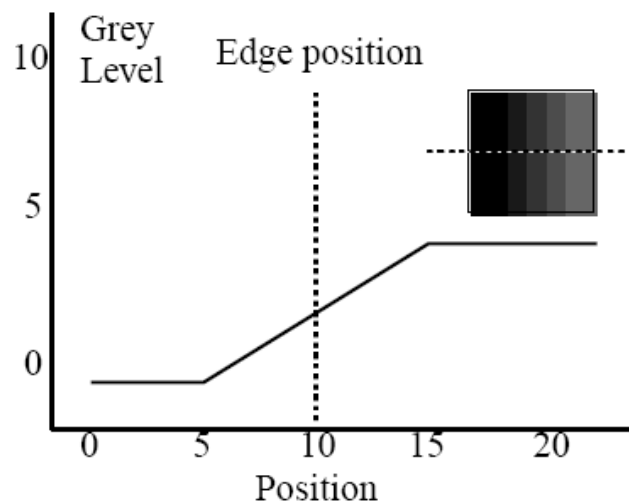
(a)



(b)



(c)



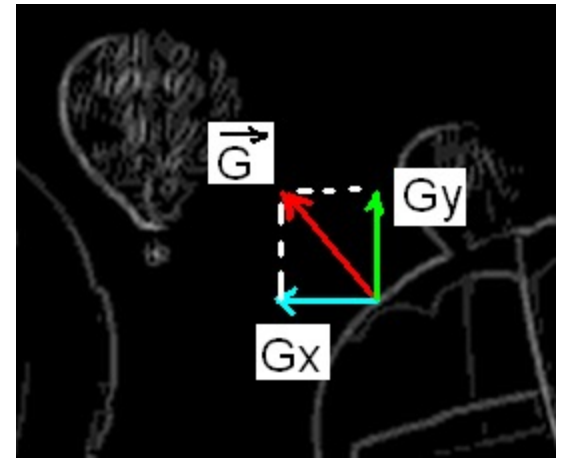
(d)



Gradientul imaginii (derivata de ordin 1)

Gradientul unei funcții 2D

$$G[f(x, y)] = \begin{bmatrix} G_{f_x} \\ G_{f_y} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$
$$= \begin{bmatrix} \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \\ \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \end{bmatrix}$$



Pentru o imagine digitală: $\Delta x = \Delta y = 1$

$$G[f[i, j]] = \begin{bmatrix} f[i + 1, j] - f[i, j] \\ f[i, j + 1] - f[i, j] \end{bmatrix} \Rightarrow \text{Convoluție cu:}$$
$$G_x = \begin{bmatrix} -1 & 1 \end{bmatrix}$$
$$G_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$



Aproximări ale gradientului

Operatorul Roberts

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$



Operatorul Prewitt

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



Operatorul Sobel

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$





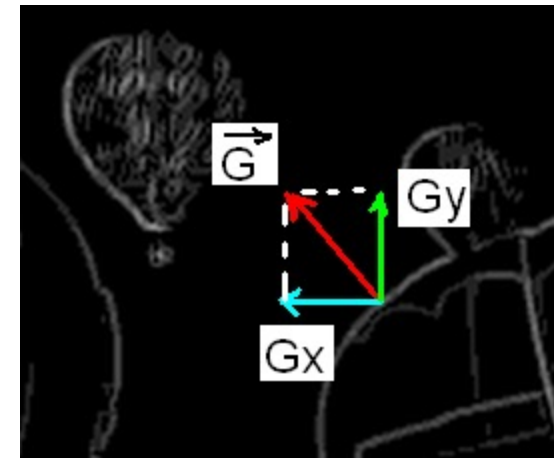
Caracteristicile gradientului

Magnitudinea

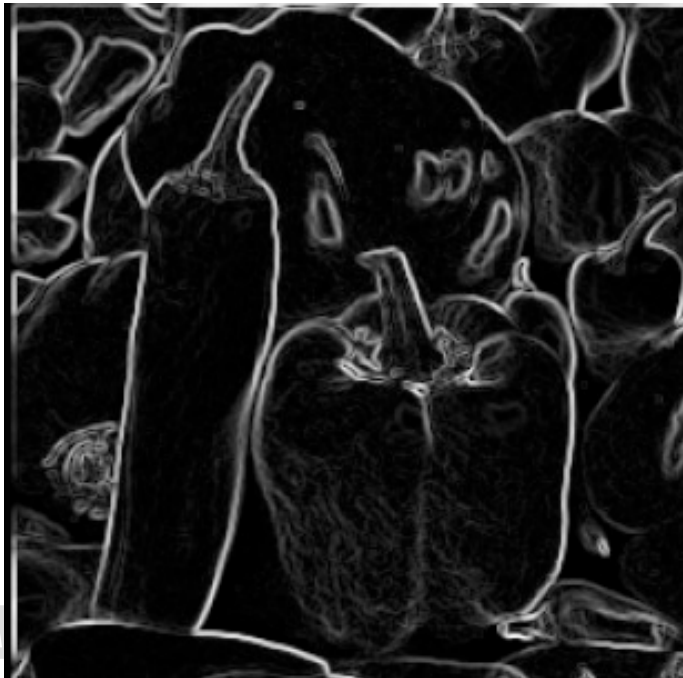
$$|G| = \sqrt{G_{fx}^2 + G_{fy}^2}$$

Direcția (orientarea)

$$dir = \arctg\left(\frac{G_{fy}}{G_{fx}}\right)$$



Imaginea $|G|$

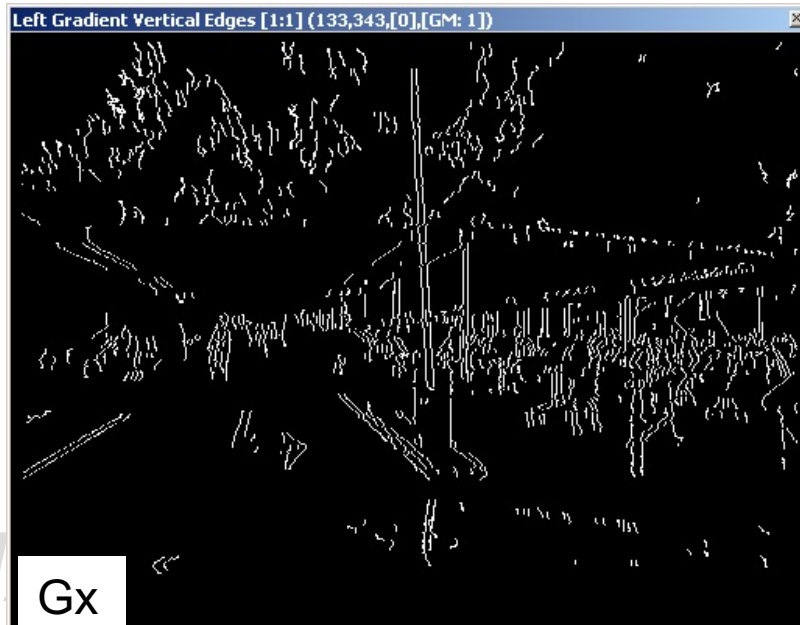


Binarizare
prag T





Exemple





Metoda Canny de detecție a muchiiilor

Caracteristici

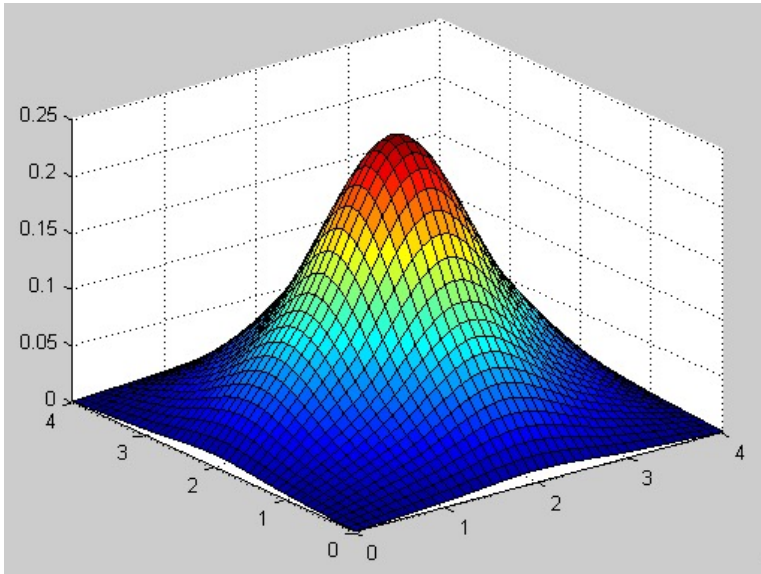
- Maximizarea raportului semnal zgomot pentru o detecție corectă
- Localizare bună la marcarea punctelor de muchie
- Minimizarea numărului de răspunsuri pozitive la o singură muchie (eliminarea non-muchiilor)

Algoritm

1. Atenuarea zgomotului folosind filtrul Gaussian
2. Calculul gradientului, modul și direcție (de exemplu, folosind operatorul Sobel)
3. Suprimarea non-maximelor
4. Binarizare cu histereză



1. Filtrare Gaussiana



$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{((x-x_0)^2 + (y-y_0)^2)}{2\sigma^2}}$$

$$g(x, y) = g(x) * g(y)$$

$$g(x) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{-\frac{(x-x_0)^2}{2 \cdot \sigma^2}}$$

$$g(y) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{-\frac{(y-y_0)^2}{2 \cdot \sigma^2}}$$

Exemplu: $\sigma = 0.8 \Rightarrow w = 5$ (latime filtrului) $w \approx 6 \cdot \sigma$

$$G(x,y) = \begin{bmatrix} 0.0005 & 0.0050 & 0.0109 & 0.0050 & 0.0005 \\ 0.0050 & 0.0521 & 0.1139 & 0.0521 & 0.0050 \\ 0.0109 & 0.1139 & 0.2487 & 0.1139 & 0.0109 \\ 0.0050 & 0.0521 & 0.1139 & 0.0521 & 0.0050 \\ 0.0005 & 0.0050 & 0.0109 & 0.0050 & 0.0005 \end{bmatrix}$$

$$f(x, y) = f_s(x, y) * g(x, y)$$



2. Calcul modul și direcție gradient

$$G[f(x, y)] = \begin{bmatrix} G_{fx} \\ G_{fy} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$G_{fx}(x, y) = f(x, y) * S_x(x, y)$$

$$G_{fy}(x, y) = f(x, y) * S_y(x, y)$$

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Magnitudinea

$$|G| = \sqrt{G_{fx}^2 + G_{fy}^2}$$

Direcția

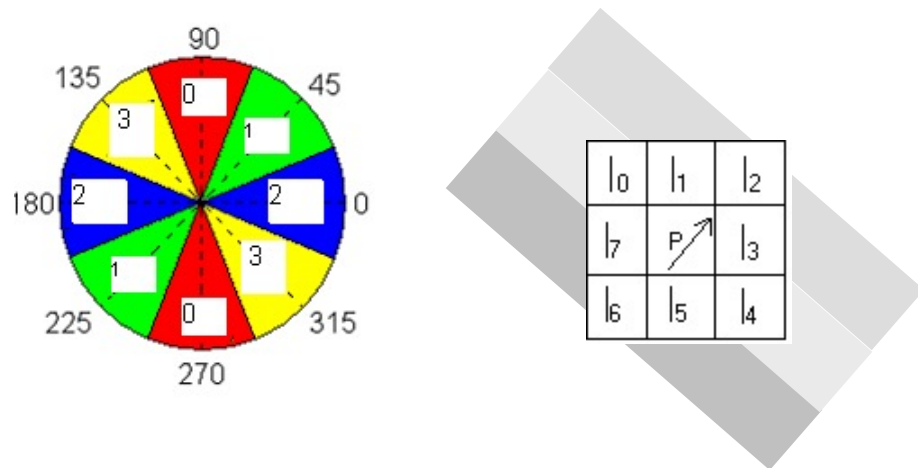
$$dir = \arctg\left(\frac{G_{fy}}{G_{fx}}\right)$$



3. Suprimarea non-maximelor

⇒ Subțierea muchiilor de-a lungul direcției gradientului (1 pixel grosime)

Cuantificarea direcției gradientului:



Exemplu: Presupunem că avem un punct P în care gradientul are direcția în zona 1.

P este un maxim local dacă:

$$G_6 \leq G \text{ \& } G_2 \leq G$$

Unde: G , G_2 , G_6 sunt magnitudinile gradientului în punctele P , I_2 , I_6 .

Dacă P este maxim local, se reține. În caz contrar, se elimină (magnitudinea lui primește valoarea zero).



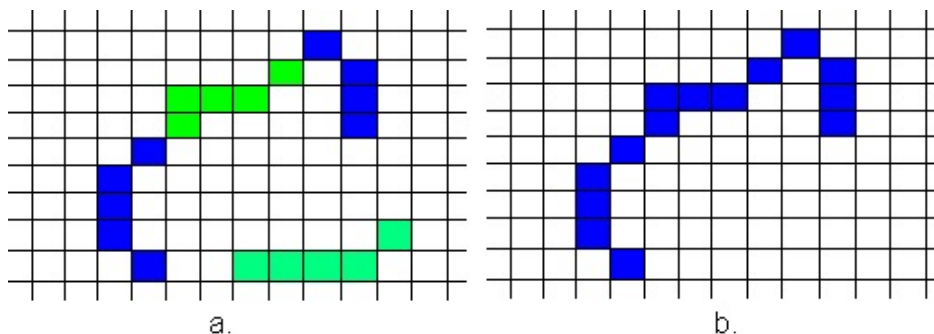
4. Binarizare cu histereză

⇒ Unirea punctelor de muchie de pe un contur (defragmentarea conturului)

1. Se folosesc două praguri: θ_L (low) și θ_H , și se aplică următoarea schemă de binarizare:

- Fiecare punct cu magnitudinea sub θ_L se etichetează ca *non-muchie* (non-edge).
- Fiecare punct cu magnitudinea peste θ_H se etichetează ca punct de *muchie-tare* (strong edge)
- Fiecare punct cu magnitudinea cuprinsă între θ_L și θ_H se etichetează ca punct de *muchie-slabă* (weak edge)

2. Se aplică un algoritm similar celui de etichetare, pentru marcarea punctelor de muchie slabă dacă acestea sunt conexe cu puncte de muchie tari, și eliminarea acelor care nu sunt conexe cu puncte de muchie tari:



a. Rezultatul după pasul 1: muchii “tari” și muchii “slabe”. b. Rezultatul după pasul 2



4. Binarizare cu histereza

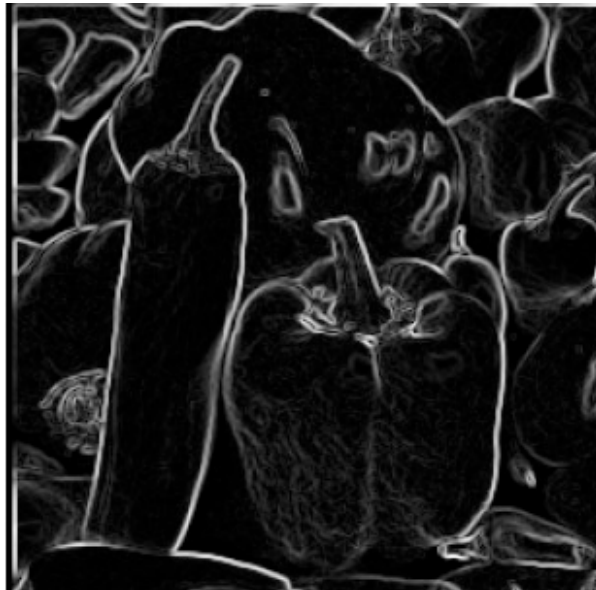
O implementare eficientă a pasului 2 folosește o coadă pentru a ține coordonatele punctelor de muchie tare. Algoritmul rulează astfel:

1. Parcurge imaginea de la stânga la dreapta și de sus în jos, găsește primul punct de MUCHIE_TARE și pune coordonatele sale în coadă.
2. Cât timp coada nu este vidă:
 - Extrage primul punct din coadă
 - Găsește toți vecinii de tip MUCHIE_SLABĂ ai acestui punct
 - Marchează în imagine vecinii acestui punct ca puncte de MUCHIE_TARE
 - Pune coordonatele acestor puncte în coadă
 - Continuă cu următorul punct din coadă
3. Continuă de la pasul 1 cu următorul punct de MUCHIE_TARE.
4. Elimină toate punctele de MUCHIE_SLABĂ din imagine, dându-le valoare NON_MUCHIE (0).



Rezultate

Imaginea $|G|$



Binarizare
prag T



Supresie
non-
maxime +
binarizare
prag fix

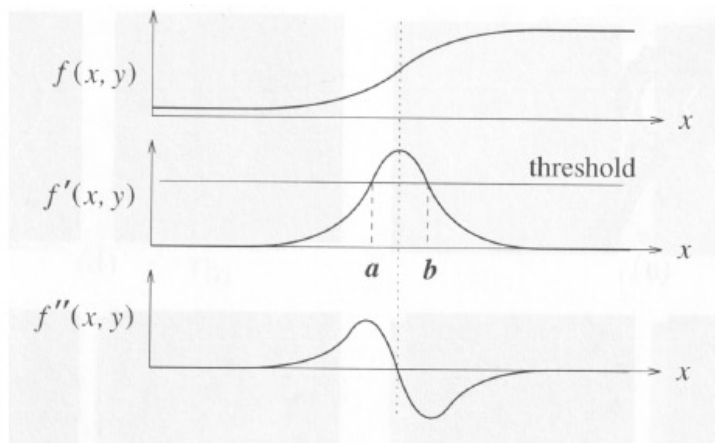


Binarizare
cu
histereza





Operatori de ordin 2 (derivata a 2-a)



$$\begin{aligned} f''(x) &= \frac{f'(x + \Delta x) - f'(x)}{\Delta x} \\ &= \frac{f(x + 2\Delta x) - 2f(x + \Delta x) + f(x)}{\Delta x^2} \end{aligned}$$

Laplacianul

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

$$\nabla^2 f(x, y) = f * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$





Operatori de ordin 2 (derivata a 2-a)

Alte forme ale Laplacianului

- prin includerea în ecuația de mai sus și a direcțiilor diagonale:

$$\nabla^2 f(x, y) = f * \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + f * \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} + f * \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix} + f * \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\nabla^2 f(x, y) = f * \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- prin utilizarea negativului Laplacianului:

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



$$\nabla^2 f(x, y) = f * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \nabla^2 f(x, y) = f * \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Laplacianul Gaussianului (LoG)

$$h(x, y) = \nabla^2[g(x, y) \otimes f(x, y)]$$

$$h(x, y) = [\nabla^2 g(x, y)] \otimes f(x, y)$$

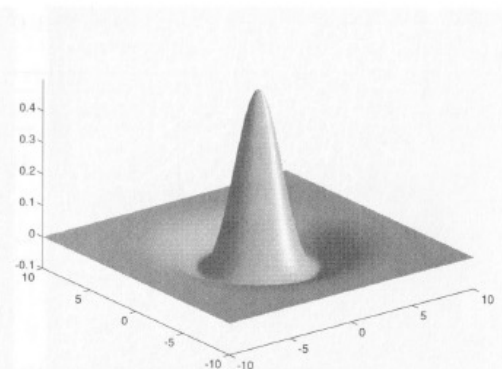
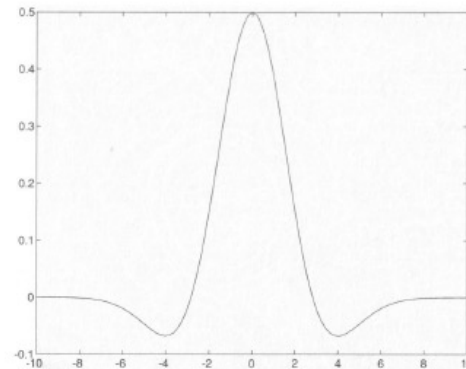


Figure 5.11: The *inverted* Laplacian of Gaussian function, $\sigma = 2$, in one and two dimensions.

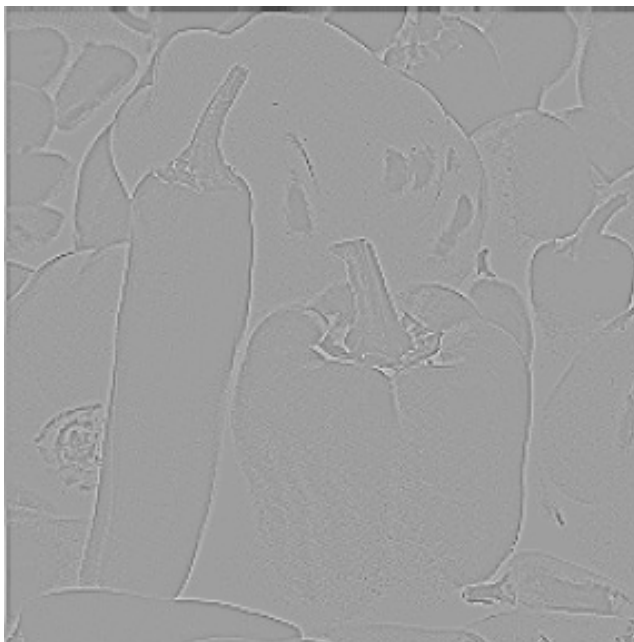
$$LoG(x, y) = \nabla^2 g(x, y) = \frac{\partial^2 g(x)}{\partial x^2} \cdot g(y) + g(x) \cdot \frac{\partial^2 g(y)}{\partial y^2}$$

$$LoG(x, y) = \frac{-1}{2 \cdot \pi \cdot \sigma^6} \cdot (\sigma^2 - x^2) \cdot e^{-\frac{x^2}{2 \cdot \sigma^2}} \cdot e^{-\frac{y^2}{2 \cdot \sigma^2}} +$$
$$+ \frac{-1}{2 \cdot \pi \cdot \sigma^6} \cdot e^{-\frac{x^2}{2 \cdot \sigma^2}} \cdot (\sigma^2 - y^2) \cdot e^{-\frac{y^2}{2 \cdot \sigma^2}}$$

$$LoG(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{2 \cdot \pi \cdot \sigma^6} \cdot e^{-\frac{x^2 + y^2}{2 \cdot \sigma^2}}$$



Laplacianul Gaussianului (LoG)



LOG compensează variația de luminozitate dintre imagini din surse diferite ale aceleiași scene

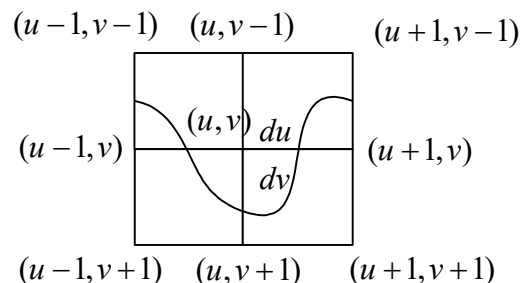
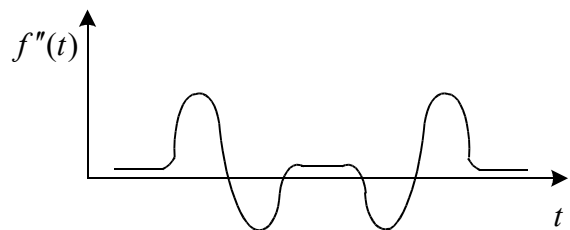
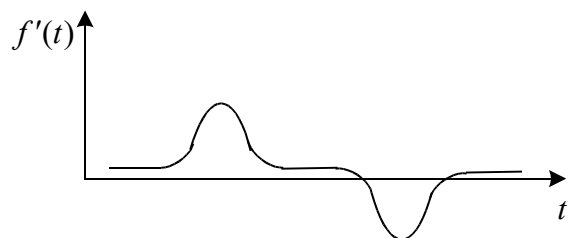
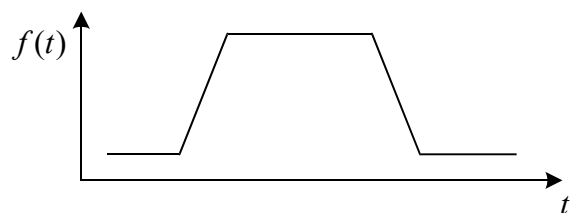
- Accentuează punctele de muchie
- Folositor pentru comparația regiunilor din imagini (exemplu: corelația stereo, optical flow)





Detecția muchiilor folosind LoG

⇒ Detecția trecerilor prin zero ale derivatei a 2-a $\{f(x,y) \otimes \text{LoG}(x,y)\}$:



Imaginea 2D a unui contur într-o vecinătate de 3x3 în jurul pixelului curent (u,v)

Determinarea poziției trecerii prin 0 a f'' la nivel de sub-pixel:

$$\text{Dacă } \text{LoG}(u,v) \cdot \text{LoG}(u+1,v) < 0: \quad du = \frac{\text{LoG}(u,v)}{\text{LoG}(u,v) + \text{LoG}(u+1,v)}$$

$$\text{Dacă } \text{LoG}(u,v) \cdot \text{LoG}(u,v+1) < 0: \quad dv = \frac{\text{LoG}(u,v)}{\text{LoG}(u,v) + \text{LoG}(u,v+1)}$$

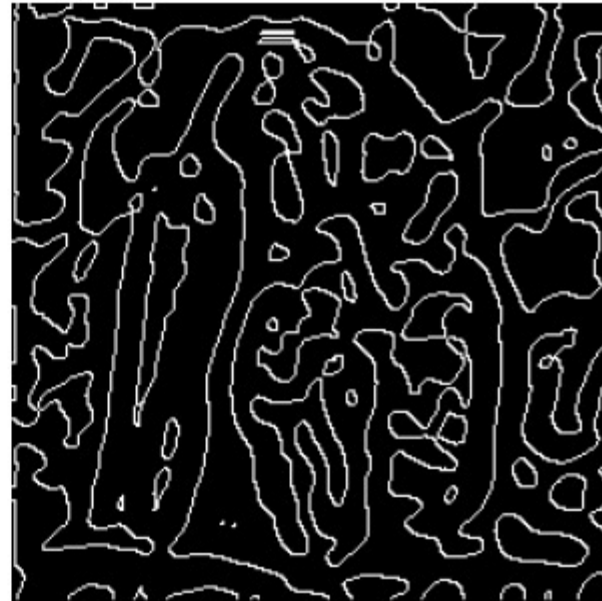


Detecția muchiiilor folosind LoG

Rezultate



$\sigma = 1.0$



$\sigma = 4.0$