



Technical University of Cluj - Napoca
Computer Science Department

Procesarea Imaginilor

Curs 10

Prelucrari pe imagini multinivel (grayscale) (V)

Detectia muchilor



La ce ne foloseste detectia de muchii ?

- Se pare ca sistemul vizual al omului foloseste muchiile ca elemente primare in procesul de recunoastere dupa care se infereaza informatii complexe legate de culoare/textura/forma
 - Este posibil sa recunoastem forme/obiecte doar pe baza conturului (ex. desene alb/negru, benzi desenate alb/negru in care este prezenta doar informatia de contur).
- ⇒ Detectia muchiilor este o etapa care poate fi folosita in analiza automata a imaginilor
- ⇒ Detectia muchiilor \approx proces de segmentare

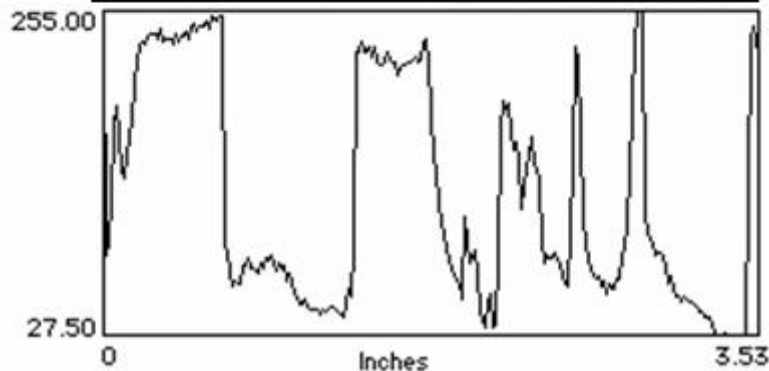
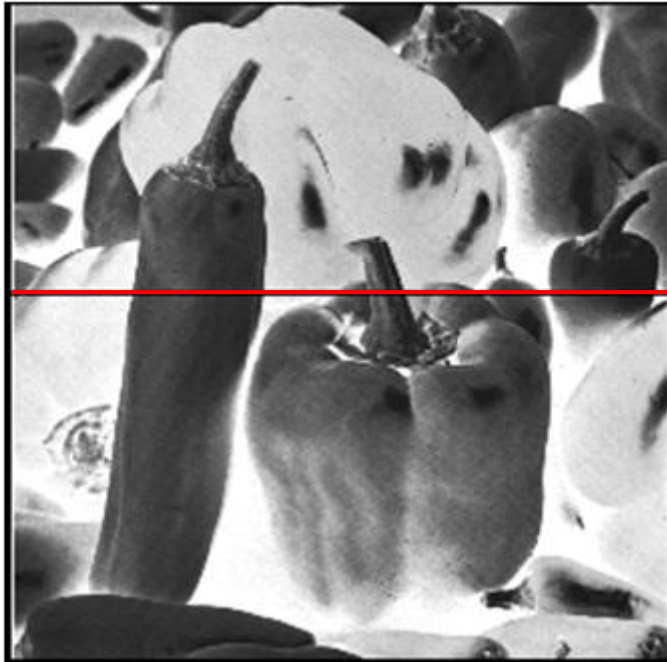
Segmentare := din informatia de nivel coborat a imaginii (pixeli / row data) ⇒ se extrage informatia de nivel ridicat:

- puncte de muchie ⇒ contur ⇒ trastauri de forma ⇒ analiza
- puncte de muchie ⇒ trasaturi pt. stereoreconstructie

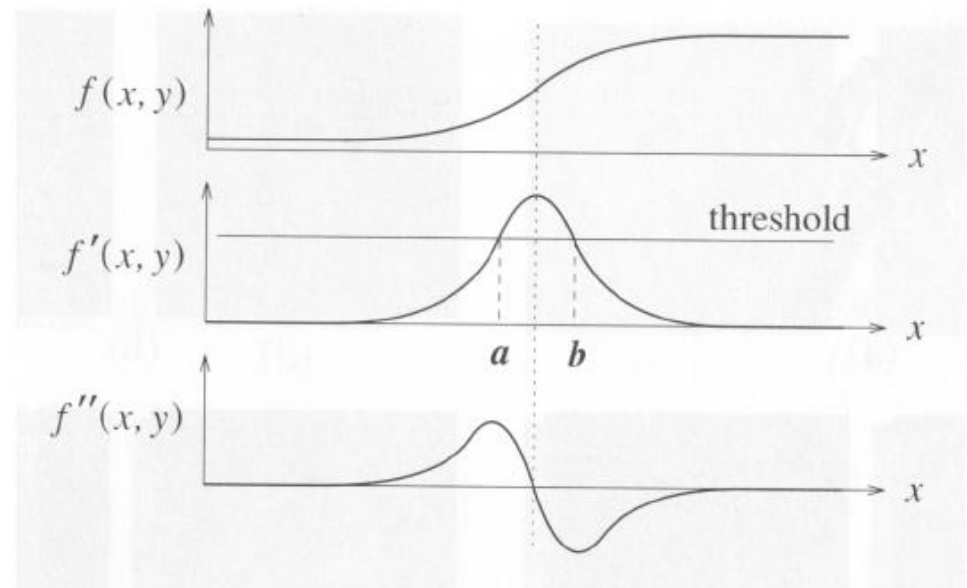


Definitie

Muchie := frontiera care separa doua regiuni de intensitate diferita (intensitatea variaza brusc intre cele doua regiuni).

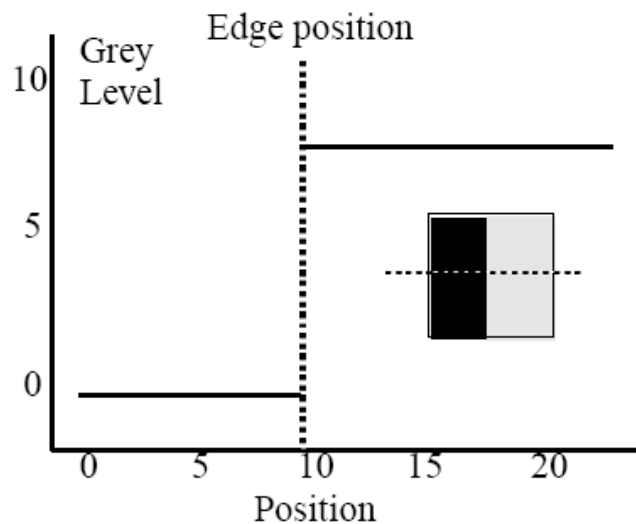


Cum putem detecta o muchie?

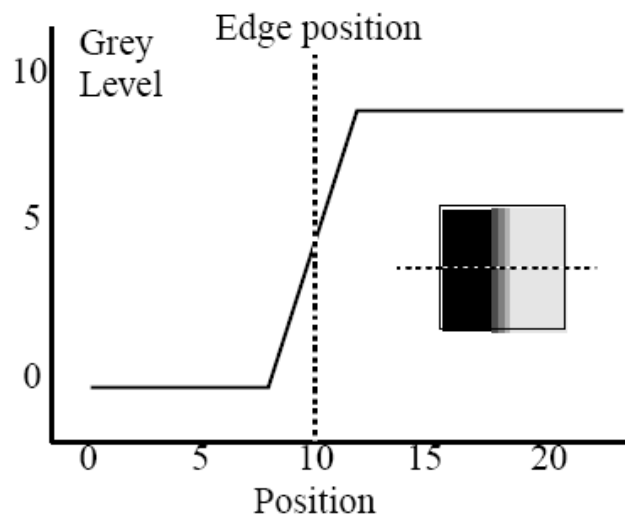




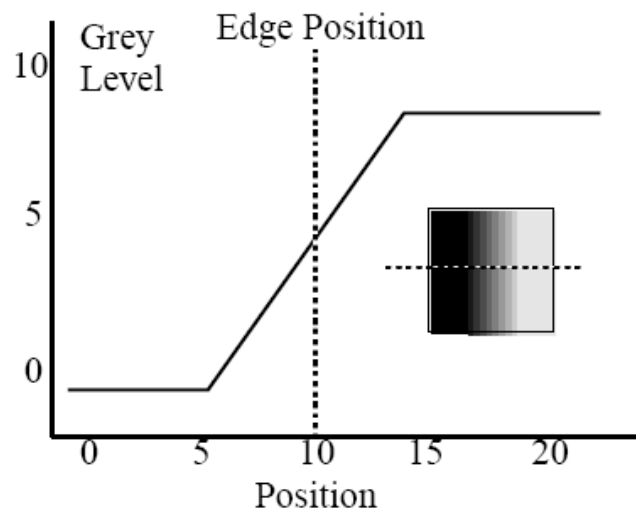
Profilul intensitatii in puntele de muchie



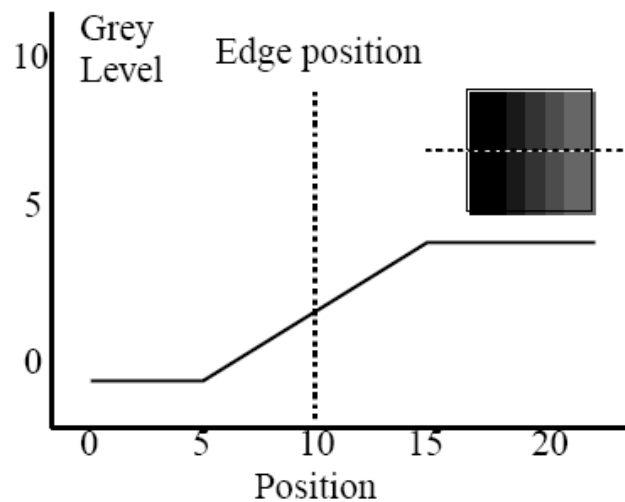
(a)



(b)



(c)



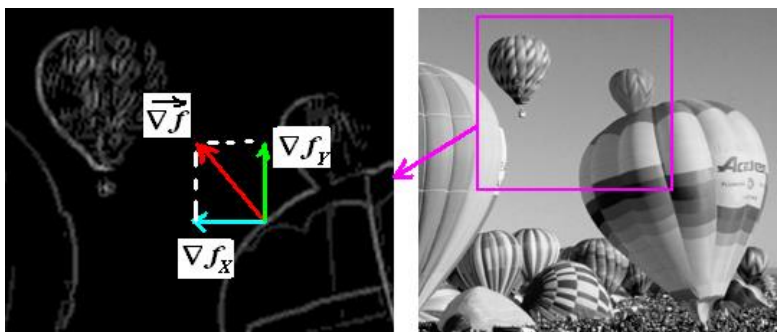
(d)



Gradientul imaginii (derivata de ordin 1)

Gradientul unei functii 2D

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



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

$$\nabla f(x, y) = \begin{bmatrix} \nabla f_x \\ \nabla f_y \end{bmatrix} = \begin{bmatrix} f[x+1, y] - f[x, y] \\ f[x, y+1] - f[x, y] \end{bmatrix} \Rightarrow \begin{aligned} \nabla f_x &= f(x, y) * \begin{bmatrix} -1 & 1 \end{bmatrix} \\ \nabla f_y &= f(x, y) * \begin{bmatrix} -1 \\ 1 \end{bmatrix} \end{aligned}$$



Aproximari ale gradientului

Operatorul Roberts

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

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

Operatorul Prewitt

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

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

Operatorul Sobel

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

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





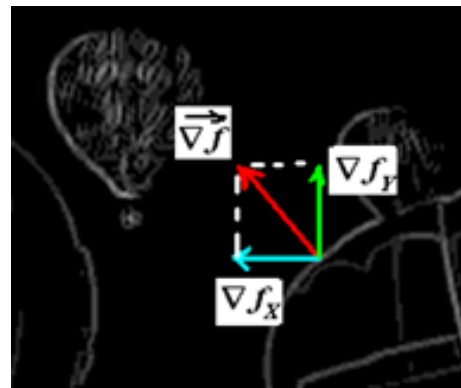
Marimi caracteristice (ale gradientului)

Magnitudinea

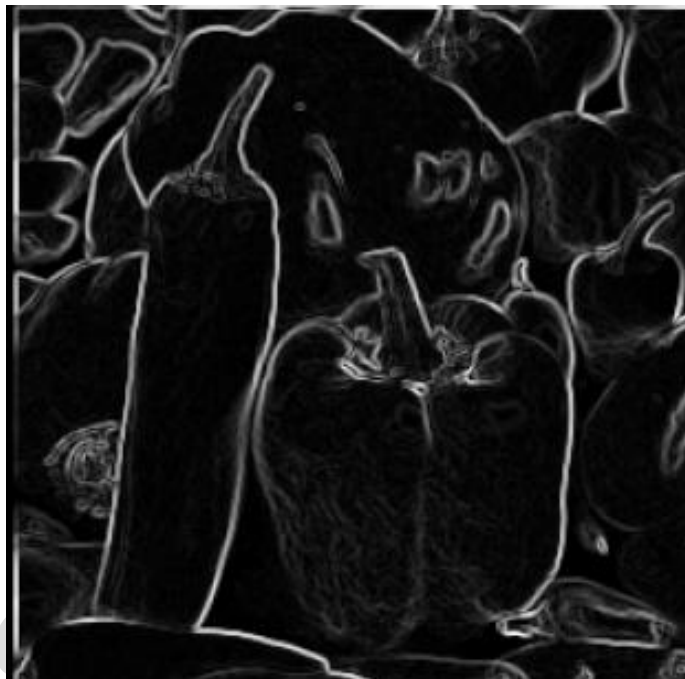
$$|\nabla f(x, y)| = \sqrt{(\nabla f_x(x, y))^2 + (\nabla f_y(x, y))^2}$$

Directia

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



Imaginea $|\nabla|$

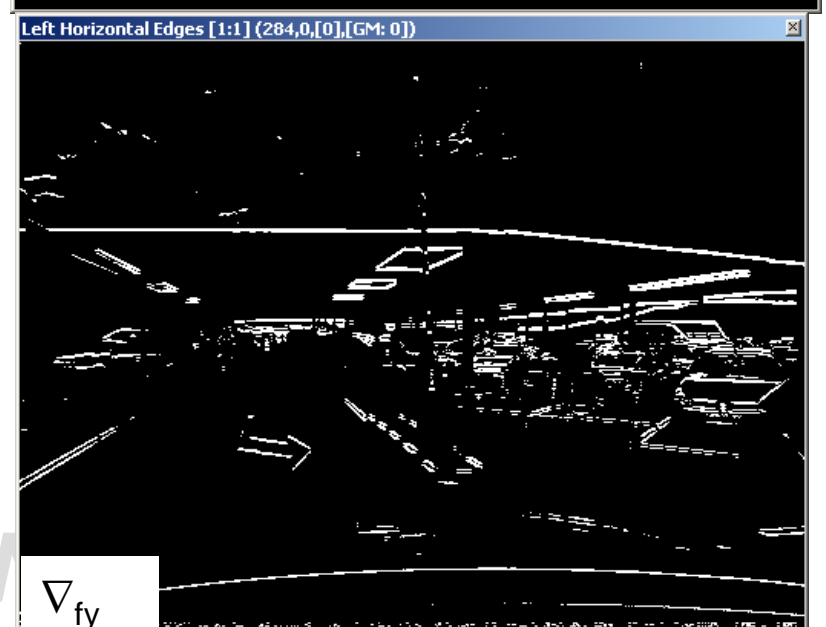
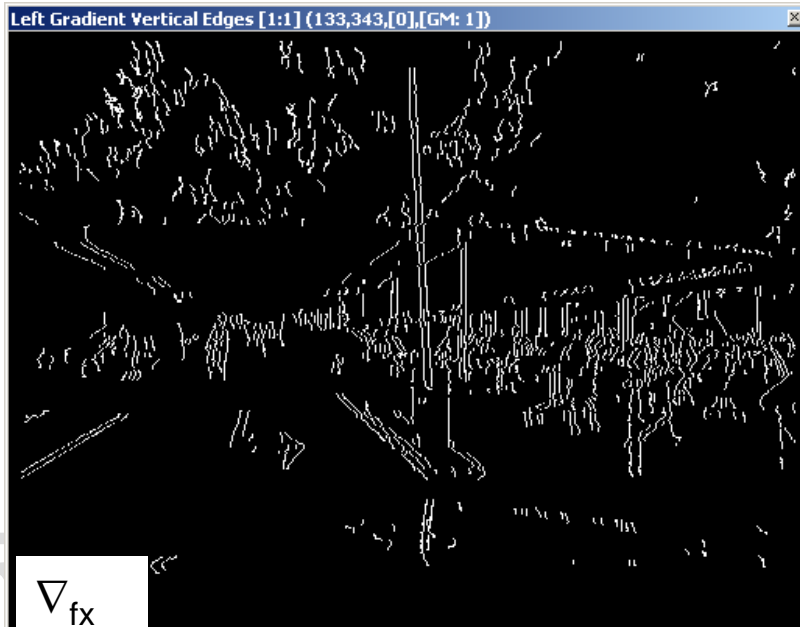
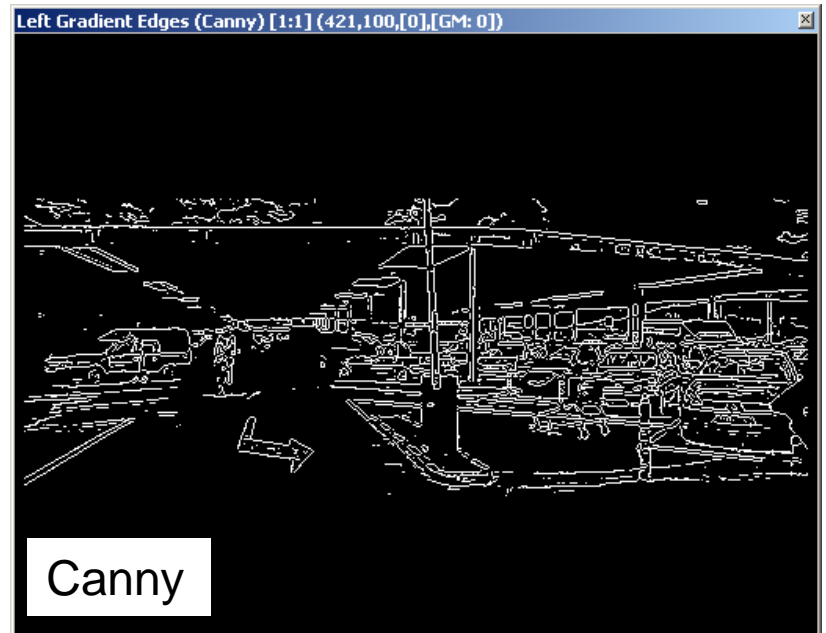


Binarizare
prag T





Exemple





Metoda Canny de detectie a muchiilor

Caracteristici

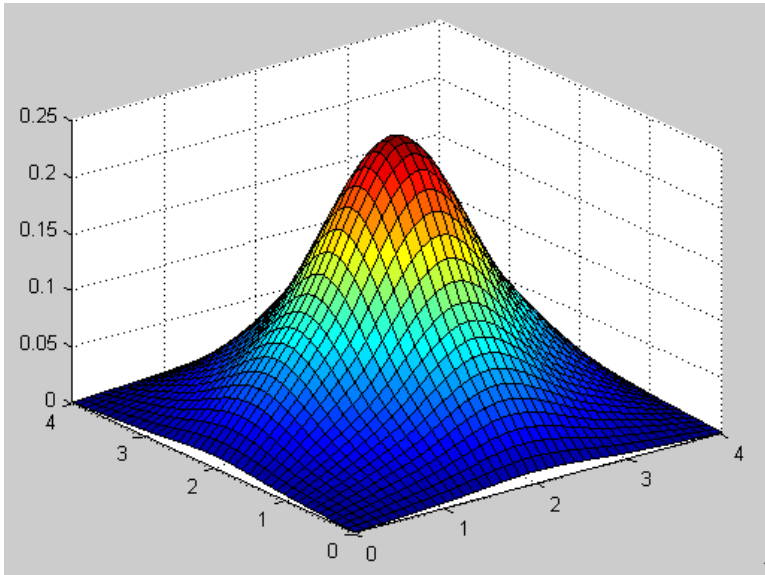
- Maximizarea raportului semnal zgomot pt. o detectie corecta.
- Localizare buna la marcarea punctelor de muchie
- Minimizarea nr. de raspunsuri pozitive la o singura muchie (eliminarea non-muchiilor)

Algoritm

1. Filtrare zgomot cu un filtru Gaussian.
2. Calculul modulului si directiei gradientului (ex. Sobel)
3. Supresia non-maximelor
4. Trhresholding cu histereza



1. Filtrare Gaussiana



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

$$g(x, y) = g(x) \cdot g(y)$$

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

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

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

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

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



2. Calcul modul si directie gradient

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

$$\nabla_{f_x}(x, y) = f(x, y) * S_x(x, y)$$

$$\nabla_{f_y}(x, y) = f(x, y) * S_y(x, y)$$

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

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

Magnitudinea

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

Directia

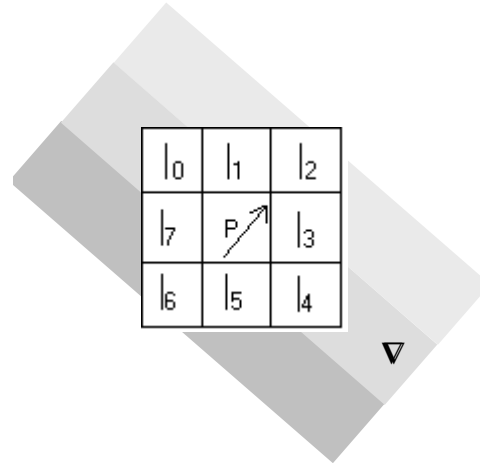
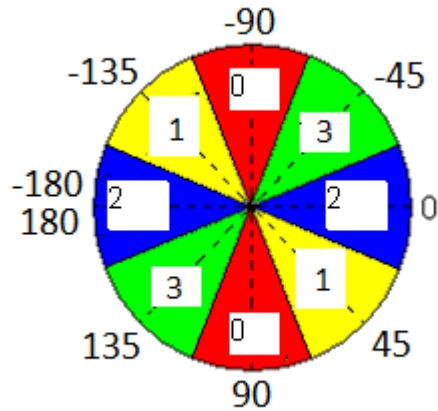
$$\theta(x, y) = \arctg\left(\frac{\nabla f_y(x, y)}{\nabla f_x(x, y)}\right)$$



3. Supresia non-maximelor

⇒ Subtierea muchiiilor de-a lungul directiei gradientului (1 pixel grosime)

Cuantificarea directiei gradientului:



Exemplu: Presupunem ca avem un punct in care gradientul are directia “3”

P este un maxim local daca:

$$|\nabla_6| < |\nabla| \text{ and } |\nabla_2| < |\nabla|$$

Unde: $|\nabla|$, $|\nabla_2|$, $|\nabla_6|$ sunt magnitudinile gradientului in P, I_2 , I_6 .

Daca P este maxim local se retine. Altfel se elimina.



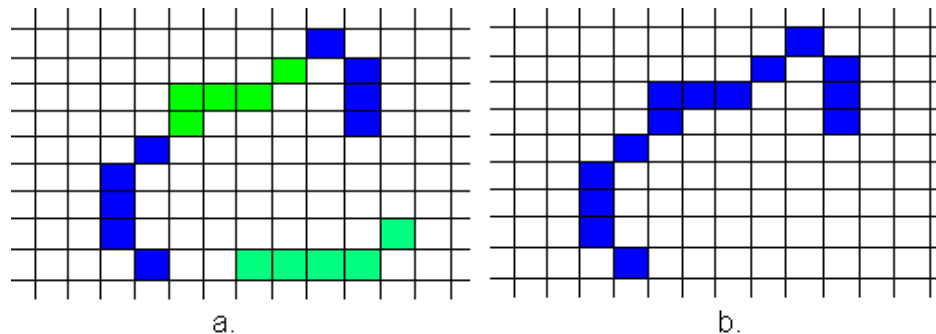
4. Thresholding cu histereza

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

1. Se folosesc doua praguri: θ_L (low) si θ_H (high) si se aplica urmatoarea schema de binarizare:

- Fiecare punct cu magnitudinea sub θ_L se eticheteaza ca *non-muchie* (non-edge).
- Fiecare punct cu magnitudinea peste θ_H se eticheteaza ca punct de *muchie-tare* (strong edge)
- Fiecare punct cu magnitudinea cuprinsa intre θ_L si θ_H se eticheteaza ca punct de *muchie-slaba* (weak edge)

2. Se aplica un algoritm similar celui de etichetare pentru marcarea punctelor de muchie slabe daca acetsea sunt conexe cu puncte de muchie tari si eliminarea acelor care nu sunt conexe cu puncte de muchie tari:



a. Rezultatul dupa pasul 1: muchii “tari” si muchii “slabe”. b. Rezultatul dupa pasul 2



4. Thresholding cu histereza

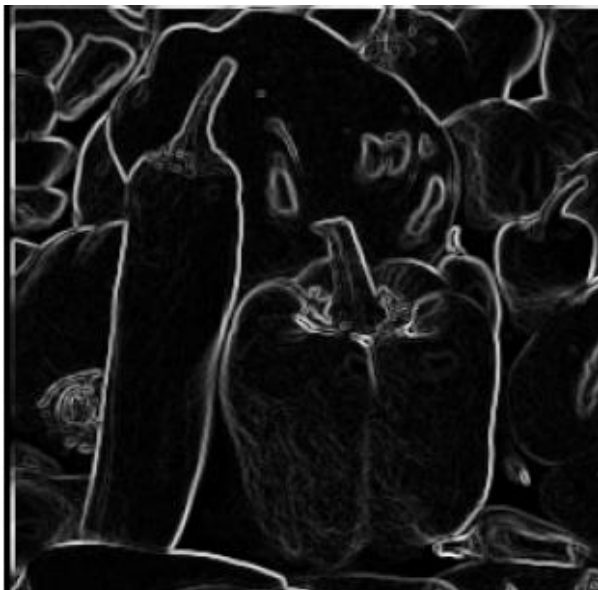
O implementare eficientă a pasului 2 folosește o coadă pentru a ține coordonatele punctelor de muchie tare. Algoritmul (BFS) rulează astfel:

1. Parcurge imaginea de la stânga la dreapta și de sus în jos și 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.



Rezultate

Imaginea $|\nabla|$



Supresia
non-
maximelor

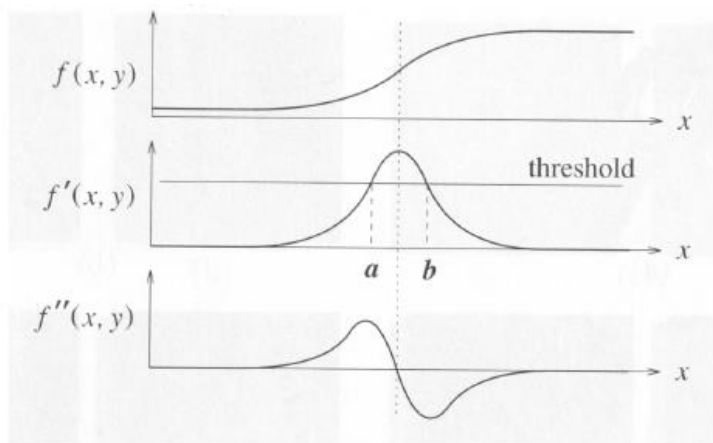


Binarizare
cu
histereza





Operatori de ordin 2 (derivata a 2-a)



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

Inlocuim in ec. de mai sus pe x cu $x - \Delta x$:

$$f_x'' = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2}$$

$$f_y'' = \frac{f(y + \Delta y) - 2f(y) + f(y - \Delta y)}{\Delta y^2}$$

Laplacianul

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



Operatori de ordin 2 (derivata a 2-a)

Alte forme ale laplacianului

- prin includerea in ec. de mai sus si a directiilor 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/ Mar-Hilderth)

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

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

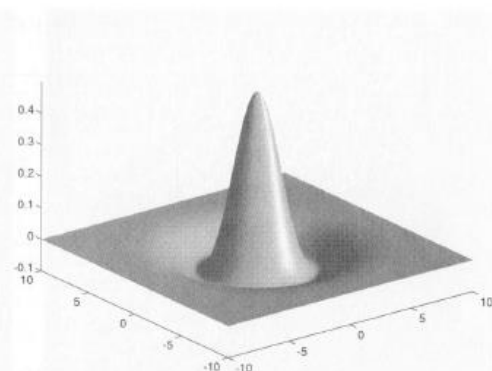
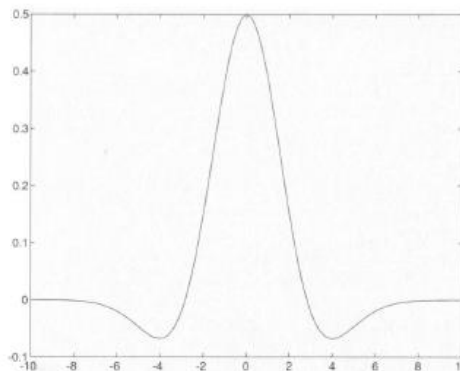


Figure 5.11: The *inverted* Laplacian of Gaussian function, $\sigma = 2$, in one and two dimensions.

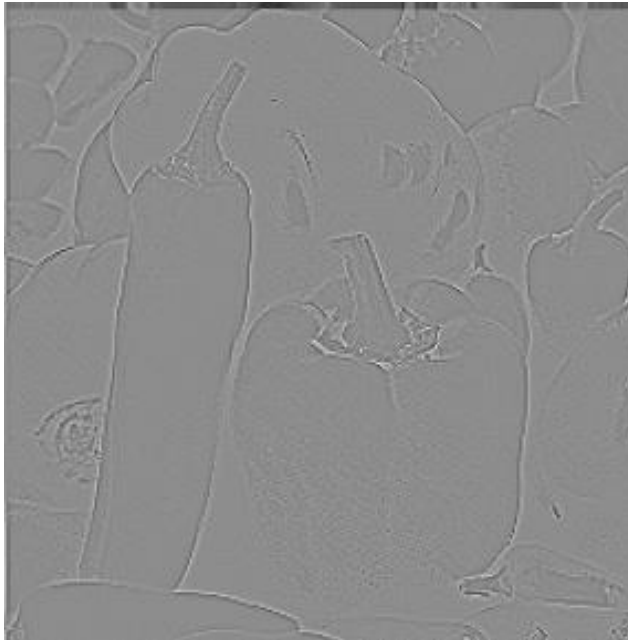
$$LoG(x, y) = \nabla^2 g(x, y) = \frac{\partial^2 g(x)}{\partial x^2} \cdot g(y) + g(x) \cdot \frac{\partial^2 g(y)}{\partial y^2}$$

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

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



Laplacianul Gaussianului (LoG)



Aplicatie: stereo corelatie (compenseaza variatia de luminozitate dintre camera stanga si dreapta).

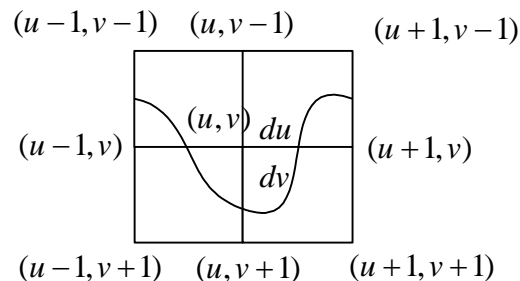
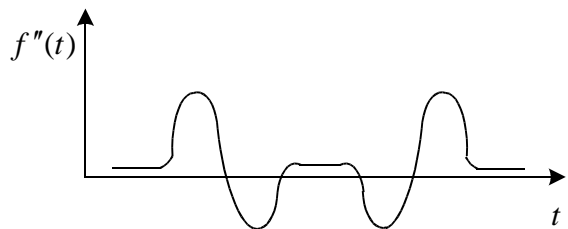
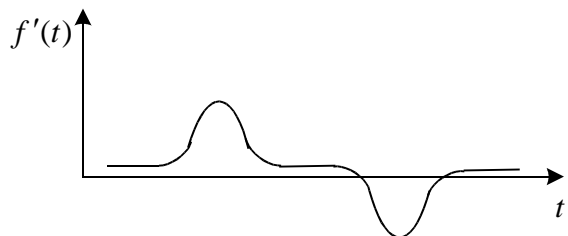
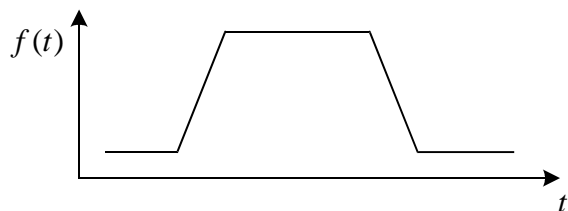




Detectia muchiiilor folosind LoG

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

Discriminarea intre trecherile prin 0 ale f'' care corespund unui max. al $|f'|$ fata de puncte de inflexiune ($f''=0$ dar $|f'|$ nu e maxim ⇒ f' schimbare de semn)



Imaginea 2D a unui contrur intr-o vecinatate de 3x3 in jurul pixelului curent (u,v)

Determinarea pozitiei trecherii prin 0 a f'' la nivel de sub-pixel:

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

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



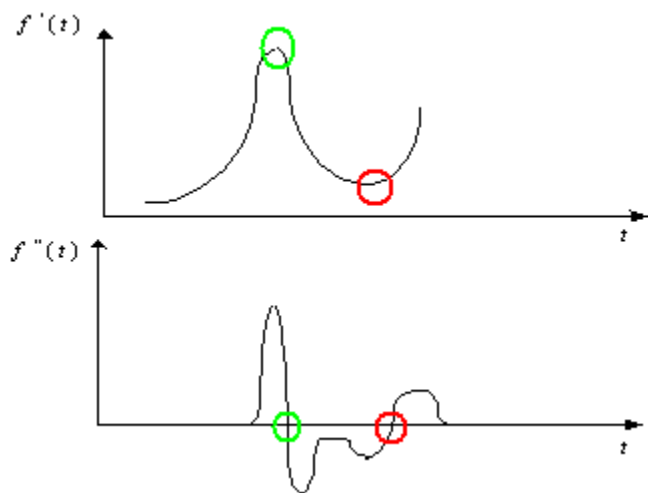
Detectia muchiiilor folosind LoG

Discriminarea trecherilor prin 0 ale f''

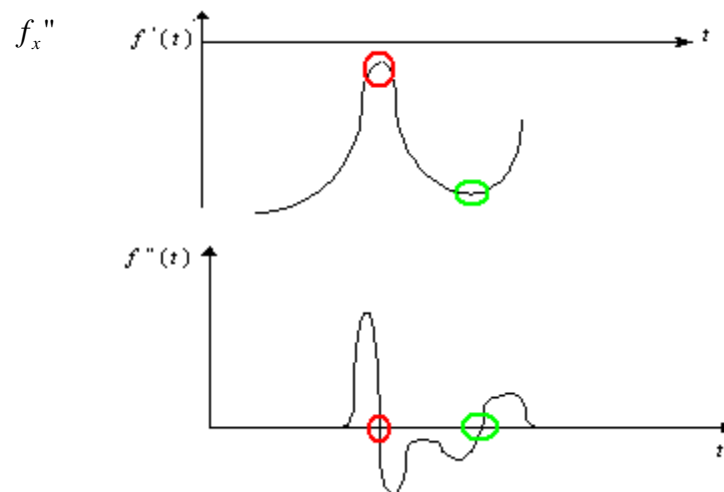
Cazul $f'(u,v) * f'(u+1,v) < 0$ si $f'(u,v) * f'(u,v+1) < 0$

\Rightarrow Daca f' are o schimbare de semn (punct de inflexiune) \Rightarrow **SE ELIMINA**

Cazul $f' > 0$



Cazul $f' < 0$



Etapă finală: thresholding cu histereza

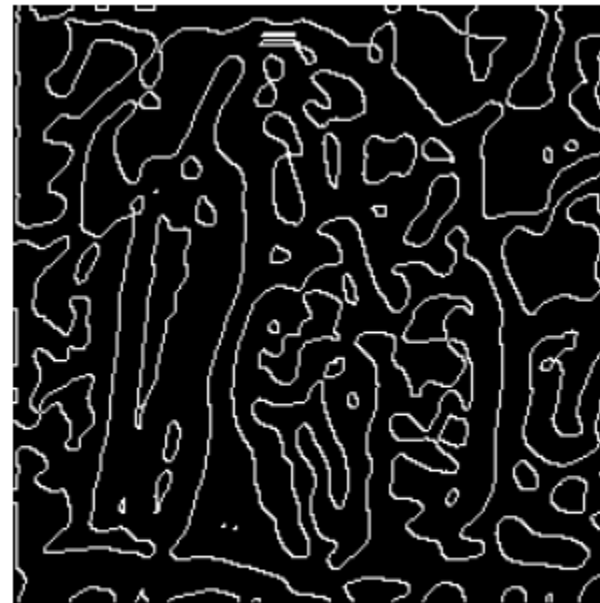


Detectia muchiiilor folosind LoG

Rezultate



sigma = 1.0



sigma = 4.0