

## Filtrarea trece-sus a imaginilor

# 1 Filtre trece sus - *High-pass Filters*

Detectarea marginilor este o etapă premergătoare pentru segmentarea bazată pe contururi. Astfel de metode de segmentare partizionează imaginea în funcție de modificări locale semnificative ale intensității de culoare. Aceste modificări sunt considerate margini între două obiecte sau între un obiect și fundal.

Trebuie făcută următoarea observație: există o diferență de semnificație între conceptul de ”margine” - *edge* și cel de contur - *contour/boundary*. Conceptul de margine este local și se referă la variații locale ale nivelelor de gri. Conceptul de contur are un înțeles global, referindu-se la conturul unui obiect/regiuni. Un contur este alcătuit dintr-o multime conectată de pixeli margine - *edge*.

Astfel există două mari problemele care se pun în cazul segmentării bazate pe contur. În primul rând se cere detectarea efectivă a variațiilor semnificative ale culorii, unde ”semnificativ” depinde de scopul urmărit și de tipul de imagine. Acest lucru se efectuează utilizând în primul rând metode de filtrare trece-sus combinate cu thresholding, ceea ce conduce adesea la contururi fragmentate, datorită componentei de zgomot sau a deficiențelor de iluminare și achiziție. Din acest motiv următoarea etapă este aceea de conectare a segmentelor de contur determinate, astfel încât să cuprindă efectiv marginile obiectelor de interes.

În cazul imaginilor în tonuri de gri, diferența de culoare se referă evident la variația tonului de gri, adică a unui scalar. În cazul imaginilor color se pune întrebarea, cum tratăm variația culorii, dat fiind că este vorba de un spațiu vectorial 3D. În cele ce urmează va fi discutată problema detectării de margini pentru imagini color, iar la finalul secțiunii se va discuta problema în cazul imaginilor RGB.

# 2 Tipuri de margini

În figura 1 sunt ilustrate 4 modele de margini care pot apărea în imagini:

- (a) Margine sub formă de treaptă - *step edge* - este un tip de margine ideală caracterizată printr-o trecere bruscă la distanță de un pixel între două tonuri de gri diferite. Asemenea margini apar în general în imaginile sintetice generate pe calculator.

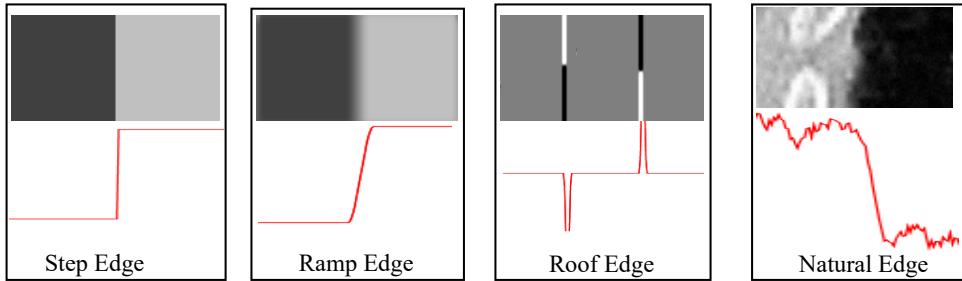


Figura 1: a) Margine treaptă - *step edge*; b) margine rampă - *ramp edge*; c) margine acoperiș - *roof edge*; d) margine realistă.

Marginile de tip treaptă sunt utilizate ca modele pentru dezvoltarea filtrelor trecesus.

- (b) Margini sub formă de rampă - *ramp edge* - reprezintă o tranziție liniară prin tonuri de gri echidistante. Asemenea contururi sunt modelate în imaginile sintetice.
- (c) Margini ”acoperiș” - *roof edge* - constau din două rampe adiacente și sunt caracteristice liniilor dintr-o imagine. Baza și panta acoperișului sunt date de lărgimea și claritatea liniei.
- (d) O margine realistă sau naturală este o margine în care tranziția de la un ton de gri la altul nu este echidistantă. Acest tip de contur este caracteristic imaginilor lumii reale. Panta unei astfel de margini este variabilă și depinde de claritatea imaginii. În practică aceste tipuri de margini sunt modelate prin margini rampă. Panta / înclinarea rampei este determinată de gradul de netezire - *blurr* al marginii.

#### *Observații:*

- Lățimea unei margini treaptă este de un pixel. Lățimea unei margini de tip rampă / naturală depinde de panta rampei, deci de gradul de netezire al imaginii.
- O margine reprezintă cum s-a spus o variație semnificativă a tonului de gri. Astfel instrumentul matematic potrivit pentru estimarea acestei variații este derivata, și anume derivata de ordinul întâi și cea de ordinul al doilea.
- Derivata de ordinul întâi poate fi folosită pentru stabilirea existenței unei margini și a orientării acesteia. Astfel de filtre se numesc filtre de tip gradient.
- Semnul celei de-a doua derivate - operatorul Laplace - indică de care parte a marginii se află pixelii de culoare închisă și de care parte cei de culoare deschisă.

### 3 Filtre de tip gradient

#### 3.1 Modelarea matematică

Filtrele de tip gradient au la bază aproximarea derivatelor parțiale de ordinul I. Pentru o funcție bidimensională  $f$ , derivata de ordinul întâi este dată prin gradient:

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

Deoarece  $f(x, y)$  este o funcție discretă, derivatele parțiale pot fi aproximăte prin diferențe finite, notate prin:

$$\begin{aligned} \frac{\partial f(x, y)}{\partial x} &\approx D_{hx}f(x, y) = \frac{\Delta_{hx}f(x, y)}{h_x} \\ \frac{\partial f(x, y)}{\partial y} &\approx D_{hy}f(x, y) = \frac{\Delta_{hy}f(x, y)}{h_y} \end{aligned}$$

în care  $h_x = \Delta x = 1$  și  $h_y = \Delta y = 1$ . Pentru  $D_{hx}f(x, y)$  există următoarele aproximări:

- diferență înainte:  $D_{hx}f(x, y) = f_x^+ = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} = f(x + 1, y) - f(x, y)$
- diferență înapoi:  $D_{hx}f(x, y) = f_x^- = \frac{f(x, y) - f(x - \Delta x, y)}{\Delta x} = f(x, y) - f(x - 1, y)$
- diferență simetrică:  $D_{hx}f(x, y) = f_x^s = \frac{f(x + \Delta x, y) - f(x - \Delta x, y)}{2\Delta x} = f(x + 1, y) - f(x - 1, y)$

Forme analoge se găsesc pentru  $D_{hy}f(x, y)$ . Acestea corespund măștile de filtrare date de  $[-1. 1]$ ,  $[-1. 1.]$  și  $[-1. 0. 1]$ , centrate pe elementul marcat cu punct. În mod similar se procedează pentru derivata după  $y$ , măștile corespunzătoare fiind verticale.

**Gradientul imaginii:** derivata după  $x$  detectează în special contururile verticale, iar derivata după  $y$  pe cele orizontale. Ele pot fi utilizate împreună prin intermediul gradientului:

$$\nabla f = \text{grad}(f) = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

cu normă  $|\nabla f| = \sqrt{f_x^2 + f_y^2}$  și direcția gradientului  $\varphi(x, y) = \arctg \left( \frac{f_y}{f_x} \right)$ .

**Norma gradientului** reprezintă ”viteza” de variație a nivelului de gri, iar **direcția gradientului** reprezintă direcția perpendiculară pe direcția de variație a conturului într-un punct dat.

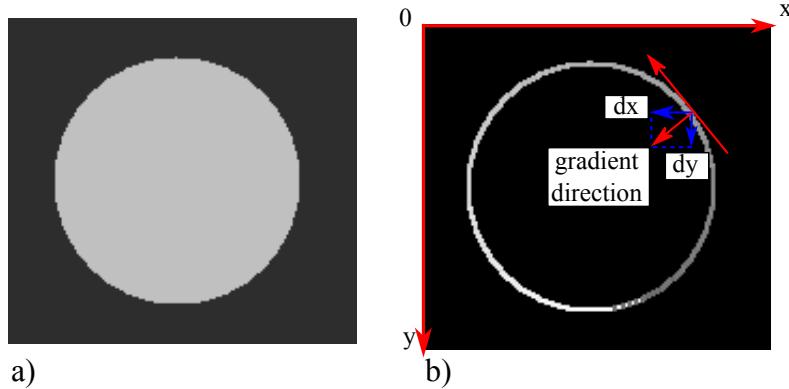


Figura 2: a) Imagine de test reprezentând un disc de culoare deschisă pe un fundal închis; b) direcția gradientului într-un punct dat al conturului care este colorat în funcție de această direcție

În figura 2 este reprezentată direcția gradientului pentru imaginea unui disc de culoare deschisă pe un fond închis. În fig. 2 b) conturul discului este colorat în funcție de direcția gradientului cu valori între [128-255] pornind de la unghiul 0 până la  $360^0$ .

Măști de filtrare pentru operatori de tip gradient sunt date în perechi: una pentru derivata după  $x$  și una pentru derivata după  $y$ . Norma poate fi aproximată prin:  $|\nabla f| = |f_x| + |f_y|$ .

### 3.2 Măști de filtrare construite pe baza gradientului

Cel mai simplu mod de implementare este prin utilizarea măștilor obținute pe baza aproximării directe a gradientului printr-una dintre cele trei tipuri de diferențe finite descrise anterior:

$$\nabla f(x, y) = (f(x+1, y) - f(x, y), f(x, y+1) - f(x, y))^T$$

$$\nabla f(x, y) = (f(x, y) - f(x-1, y), f(x, y) - f(x, y-1))^T$$

sau

$$\nabla f(x, y) = (f(x+1, y) - f(x-1, y), f(x, y+1) - f(x, y-1))^T$$

sau prin operatorul ***Roberts cross-gradient***:

$$\nabla f(x, y) = (f(x+1, y+1) - f(x, y), f(x, y+1) - f(x+1, y))^T.$$

Perechile de măști corespunzătoare acestor operatori sunt prezentate în fig. 3.

Pentru aplicații practice se preferă însă măști pătrate de dimensiuni impare  $(2l+1) \times (2l+1)$ , astfel încât să poată fi centrate asupra unui anumit pixel. Pentru  $l = 1$  aproximând derivata de ordinul întâi a imaginii  $f(x, y)$  prin diferențele simetrice  $f_x^s$  și  $f_y^s$  și extinzând empiric derivele în raport cu o anumită lungime a conturului se obțin operatorii Prewitt și Sobel, prezentate în continuare. Operatorul Sobel este preferat operatorului Prewitt,

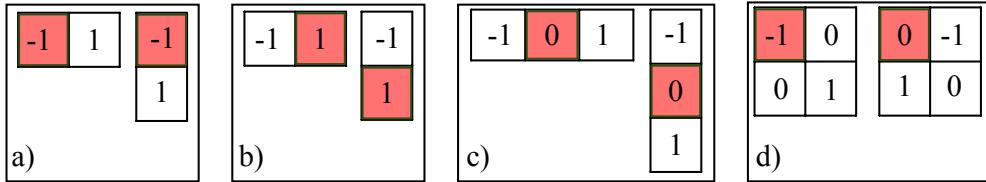


Figura 3: Măștile corespunzătoare a) diferenței înainte; b) diferenței înapoi; c) diferenței simetrice; d) Operatorului *Roberts cross-gradient*. Pixelul modificat este cel marcat cu roșu.

deoarece atribuie o pondere mai mare a vecinilor orizontali și verticali.

### Operatorul Prewitt

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

unde

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

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

### Operatorul Sobel

Pentru a reduce sensibilitatea la zgomot a acestui operator, vecinilor verticali și orizontali ai pixelului central li se asociază ponderea 2, obținându-se astfel filtrul Sobel. În acest caz gradientul poate fi aproximat prin

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

unde

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

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

Măștile de filtrare Prewitt și Sobel în direcția  $x$  respectiv  $y$  sunt prezentate în figura 4.

**Observații:** Măștile  $P_x$  și  $S_x$  din fig. 4 evidențiază în mod predilect contururile orizontale, iar măștile  $P_y$  și  $S_y$  din fig. 4 contururile verticale.

Datorită faptului că în măștile de filtrare trece jos există și coeficienți negativi, este posibil ca pentru anumiți pixeli să producă valori negative. Aceste valori nu sunt reprezentabile

a)	$P_x$	b)	$P_y$
-1	0	-1	-1
-1	0	1	0
-1	0	1	1

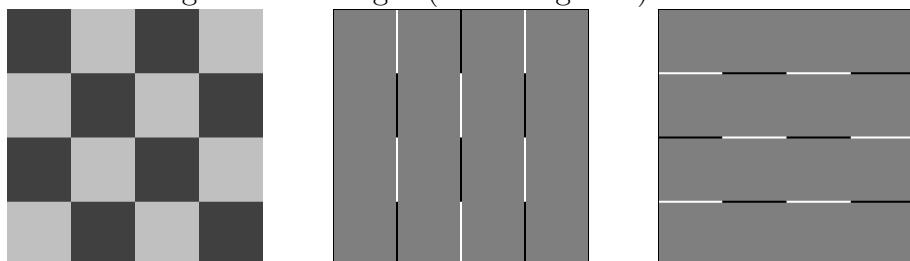
-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

Figura 4: Măștile corespunzătoare a) operatorului Prewitt; b) operatorului Sobel.

pe ecran. De fapt, pentru un pixel  $(x, y)$  aflat la trecerea de la o regiune închisă la una deschisă valoarea produsă de masca Sobel respectiv Prewitt va fi pozitivă, iar pentru trecerea de la închis la deschis, valoarea va fi negativă. Dacă se dorește punerea în evidență a acestor valori, se poate realiza o translatare cu 127.5 a valorilor produse de mască și reprezinta imaginea rezultată. Atunci zonele uniforme vor fi de culoare gri (acolo valoarea produsă de mască este aproape de 0, dar a fost translatată cu 127.5), valorile negative vor fi colorate cu tonuri de gri închise spre negru, iar valorile pozitive cu tonuri deschise spre alb. O astfel de reprezentare poate fi văzută în exemplele următoare.



**Exemplu:** de imagine filtrată cu filtrele  $S_x$  și  $S_y$  separat pentru o imagine sintetică. Se observă că filtrul  $S_x$  (imaginea din mijloc) a detectat marginile verticale, iar filtru  $S_y$  (imaginea din dreapta) a detectat marginile orizontale. Acolo unde se realizează trecerea de la un pătrat închis la unul deschis, al modelului de săh, marginea produsă este albă (valori pozitive produse de mască), iar acolo unde se face trecerea de la un pătrat închis la unul deschis marginea este neagră (valori negative).



**Exemplu:** de imagine filtrată cu filtrele  $S_x$  și  $S_y$  separat pentru o imagine naturală. Rezultatul filtrului  $S_x$  (imaginea din mijloc) și al filtru  $S_y$  (imaginea din dreapta) translatare cu 127.5.



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

$R_1$

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

$R_2$

Figura 5: Măștile corespunzătoare operatorului Roberts.

În practică, pentru a obține atât marginile preponderent verticale cât și pe cele preponderent orizontale se combină perechile de măști în modul următor:

- Se filtrează imaginea originală dată prin funcția discretă  $f(x, y)$  pe rând cu măștile  $P_x$  (respectiv  $S_x$ ) și  $P_y$  (respectiv  $S_y$ ) obținându-se imaginile  $g_x$  și  $g_y$ .
- Se calculează imaginea gradient.
- Dacă este de interes direcția marginilor, se calculează și aceasta.
- Se aplică un prag - *threshold* - pentru a obține contururile date prin variațiile semnificative ale nivelelor de gri.



#### To Do:

- Implementați unul dintre filtrele bazate pe gradient (Prewitt, Sobel, Roberts) în *framework*-ul de laborator.
- Căutați în documentația online funcțiile de filtrare din OpenCV pentru filtrele bazate pe gradient. Studiați care este semnificația parametrilor acestor funcții.

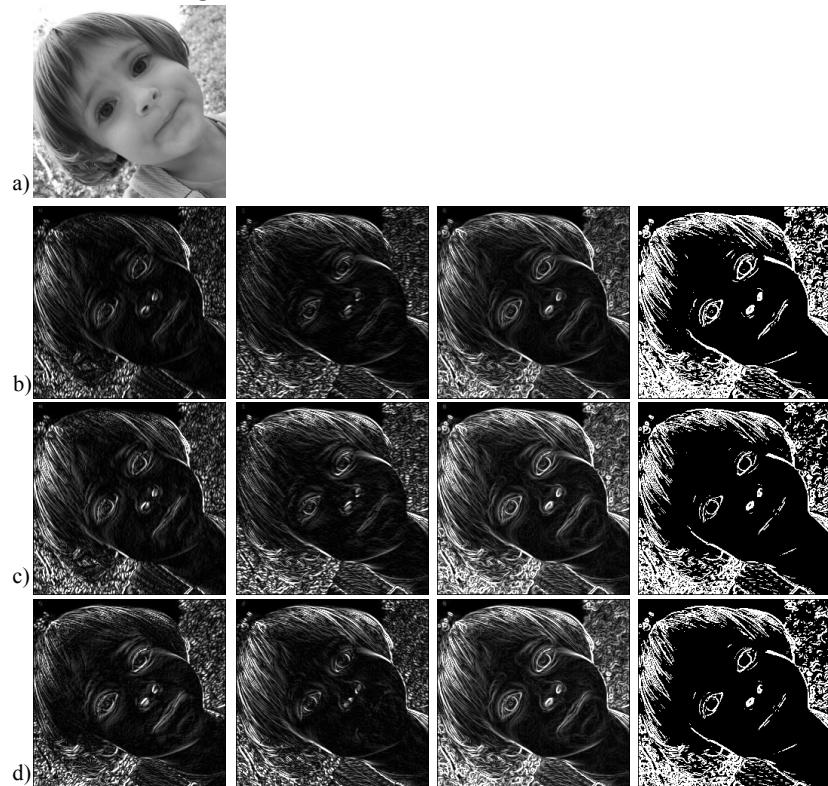
#### Observații:

- 1) Prin aplicarea unui operator de derivare se obțin informații despre contur fig. ???. Pentru a obține o imagine binară conținând doar puncte de contur trebuie în plus aplicat un prag care stabilește, care dintre valorile nenule obținute din diferențe reprezintă un contur efectiv și care dintre acestea sunt considerate doar variații nesemnificative în interiorul unei regiuni relativ uniforme. Această operație se numește *thresholding*.
- 2) Derivata de ordinul întâi produce contururi mai groase, acolo unde tranziția între două nivele de gri se face printr-o rampă mai puțin abruptă, iar pentru realizarea imaginii binare conținând punctele de contur se consideră valoarea absolută a diferenței dată de masca de filtrare.
- 3) Direcția gradientului este ortogonală pe direcția conturului în punctul dat.

Pentru detectarea contururilor diagonale pot fi utilizate măștile de filtrare ale operatorului Roberts, prezentate în fig. 5.



**Exemplu** de aplicare a filtrelor Prewitt, Sobel și Roberts pe o imagine digitală în tonuri de gri.



a) Imagine originală de test; Contururi orizontale și verticale, imaginea gradient și imaginea binară obținută după aplicarea unui prag  $t = 150$  asupra imaginii gradient obținute cu b) Filtrul Prewitt; c) Filtrul Sobel; d) Contururi diagonale și imaginea gradient obținute prin filtrarea cu filtrul Roberts.

### 3.3 Operatorul Canny

Problema principală care apare la detectarea de contur cu operatorii prezenți - Sobel, Prewitt, Roberts - este obținerea de contururi fragmentate și mai groase decât un pixel. Acest lucru se datorează pe de-o parte prezenței zgomotului în imagine, a iluminării neuniforme și a faptului că în imagini reale nu există margini treaptă, care ar produce contururi subțiri.

Operatorul Canny a fost dezvoltat pentru:

- reducerea ratei de eroare în detectarea pixelilor de contur;
- localizarea corectă a conturului detectat, cât mai aproape de conturul real;

- obținerea unui singur pixel de contur, corespunzător unui punct de pe conturul real.

Cele patru etape principale ale operatorului Canny sunt:

1. Netezirea - *Smoothing*: pentru reducerea zgomotului și deci reducerea apariției de margini false.
2. Detectarea marginilor - *High-pass filtering*: prin filtrare trece-sus, pentru obținerea informației de contur.
3. Eliminarea valorilor nemaximale pentru margini - *Non-maxima suppression*: pentru obținerea de contururi de grosime 1 pixel.
4. Prăguire întârziată - *Hysteresis thresholding*: pentru eliminarea marginilor care nu sunt considerate contururi și obținerea de contururi nefragmentate.

### 3.3.1 Netezirea

Pentru netezirea imaginii în cazul operatorului Canny se utilizează de obicei filtrul Gaussian. Dimensiunea filtrului influențează gradul de netezire, deci pe de-o parte cât de mult de reduce zgomotul, dar pe de altă parte cât de exact sunt detectate contururile. Din experiență recomandăm o dimensiune de  $5 \times 5$  sau  $7 \times 7$  a măștii de filtrare ( $\sigma \in [1, 2]$ ).

### 3.3.2 Detectarea marginilor

Detectarea marginilor poate fi realizată printr-un operator de gradient. De obicei se utilizează un filtru Sobel.

### 3.3.3 Non-maxima suppression

Prin netezire, urmată de aplicarea filtrului Sobel se obțin margini, de obicei mai groase de un pixel. Ideea din spatele etapei de *Non-maxima suppression* este aceea de a subția marginile astfel obținute. Pentru a determina direcția de subțiere a marginilor, este necesară estimarea orientării acestora. Orientarea unei margini poate fi determinată pe baza gradientului în punctul respectiv, prin calcularea unghiului gradientului cu formula:

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

unde  $f_y$  și  $f_x$  reprezintă derivatele după direcția verticală și respectiv orizontală. Acest unghi poate avea orice valoare în intervalul  $[-2\pi, 2\pi]$ . Este însă suficient să reducem orientarea gradientului la patru direcții principale:

- orizontală  $[-\pi/8, \pi/8] \cup [-7\pi/8, 7\pi/8]$ ,
- verticală  $[-5\pi/8 - 3\pi/8] \cup [3\pi/8, 5\pi/8]$ ,
- diagonală 1  $[-7\pi/8, -5\pi/8] \cup [\pi/8, 3\pi/8]$ ,
- diagonală 2  $[-3\pi/8, -\pi/8] \cup [5\pi/8, 7\pi/8]$ .

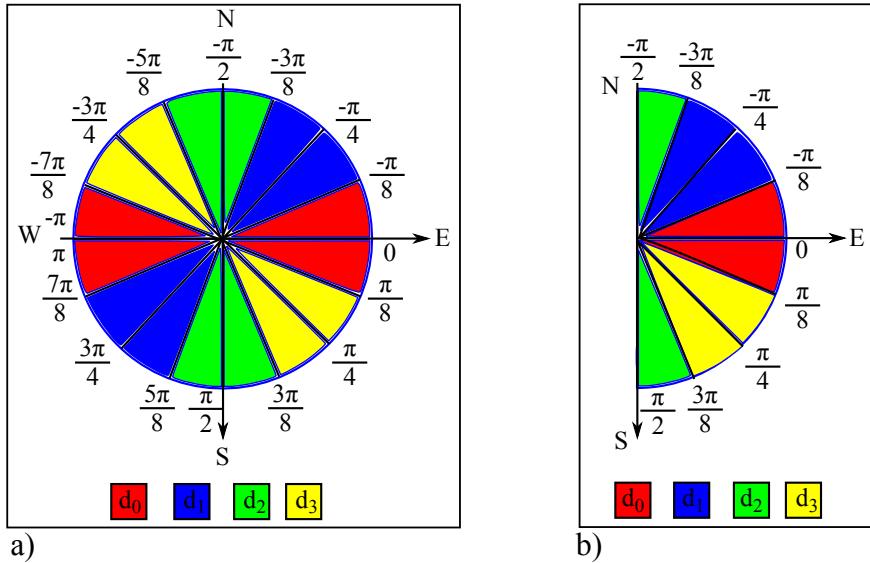


Figura 6: Reducerea direcțiilor gradientului la 4 luanăd în considerare unghiurile a) în intervalul  $[-\pi, \pi]$ , b) în intervalul  $[-\pi/2, \pi/2]$ .

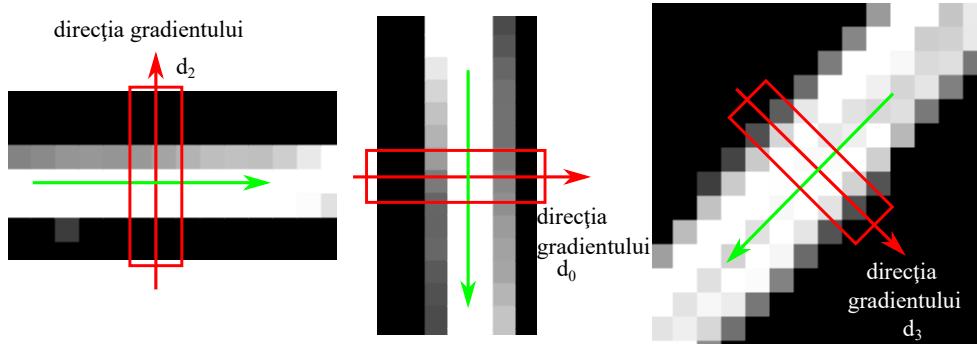


Figura 7: Directiile de subțiere pentru contururi orizontale, verticale și diagonale.

Acest lucru este ilustrat în figura 6.

Pentru fiecare pixel  $(x, y)$  se calculează  $f_x$ ,  $f_y$  utilizând măștile Sobel corespunzătoare, se determină unghiul  $\varphi(x, y)$  conform formulei anterioare, iar apoi, pe baza intervalelor de discretizare, se estimează direcția gradientului:  $d_0$  - orizontal,  $d_2$  - vertical,  $d_1$  - diagonala 1,  $d_3$  - diagonala 2.

Direcția gradientului este perpendiculară pe direcția conturului, așa cum este ilustrat în figura 7. Astfel încât, de exemplu atunci când direcția gradientului pentru un punct  $(x, y)$  este  $d_0$ , rezultă că în acel punct conturul are direcția verticală, deci se ia în considerare o vecinătate orizontală din care se păstrează doar valoarea cu norma gradientului maximă.

În figura 8 este ilustrat modul de alegere a vecinătăților pentru cele 4 direcții considerate. Se pot desigur considera și vecinătăți de câte 5 pixeli orientate la fel ca cele din figură.

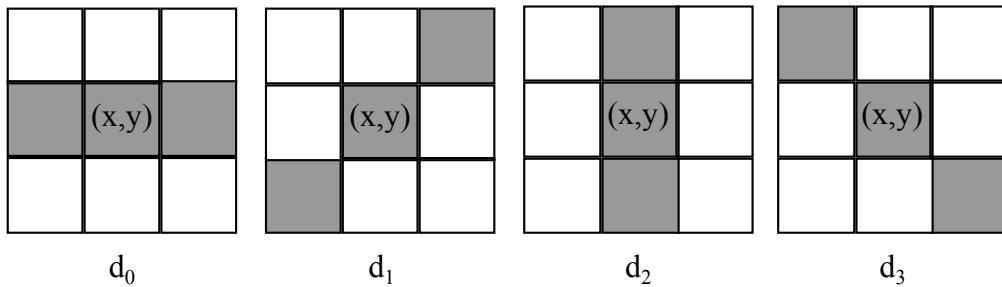


Figura 8: Vecinătățile de subțiere pentru cele 4 direcții considerate

### 3.3.4 Hysteresis thresholding

Această etapă are ca scop reducerea fragmentării contururilor, obținută în cazul utilizării unui prag dat  $T$  asupra normei gradientului, fragmentare datorată zgomotului prezent în imagine, contrastului redus în anumite zone sau calității imaginii în general.

Ideea este aceea de a alege două praguri  $T_{min}$  și  $T_{max}$ . Toți pixelii în care norma gradientului este mai mică decât  $T_{min}$  se consideră ca fiind parte din fundal, iar toți pixelii pentru care norma gradientului este mai mare decât  $T_{max}$  sunt considerați ca fiind siguri pixeli de contur.

Pixelii pentru care norma gradientului este în intervalul  $[T_{min}, T_{max}]$  sunt considerați potențiali pixeli de conturi. Ei vor fi marcați ca și pixeli de contur, doar dacă sunt conectați de un pixel cu norma gradientului mai mare decât pragul  $T_{max}$ .

Alegerea celor două praguri se realizează empiric și depinde de tipul de imagini analizate.



**Exemplu** al aplicării operatorului Canny asupra unei imagini în care iluminarea nu este uniformă. Se observă faptul că în figura b), obținută prin filtrarea imaginii din fig. a) cu un filtru Sobel, pentru care s-a utilizat pragul  $T = 100$  nu are în partea laterală a clădirii decât fragmente de contur, iar în fig. c), obținută prin aplicarea operatorului Canny cu pragurile  $T_{min} = 40$  și  $T_{max} = 60$  se obțin pe partea laterală a clădirii toate contururile determinate de către ferestre.



**To Do:** Căutați în documentația online funcția OpenCv pentru operatorul Canny. Studiați care este semnificația parametrilor acestor funcții.

## 4 Filtre bazate pe derivata de ordinul 2 - Filtrul Laplace

### 4.1 Modelare matematică

Aproximarea derivatei de ordinul doi cu  $\Delta x = 1$  se calculează pe baza aproximării derivatei de ordinul întâi, utilizând atât diferența înainte cât și diferența înapoi:

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

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

Pentru derivata după direcția  $y$  aproximarea se calculează în mod analog. În general pentru detectarea de contur - în special pentru drepte și puncte - se combină cele două derivate parțiale de ordinul doi, utilizându-se operatorul Laplace:

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

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

Măști de filtrare corespunzătoare operatorului Laplace sunt reprezentate în figura 9 a) și b).

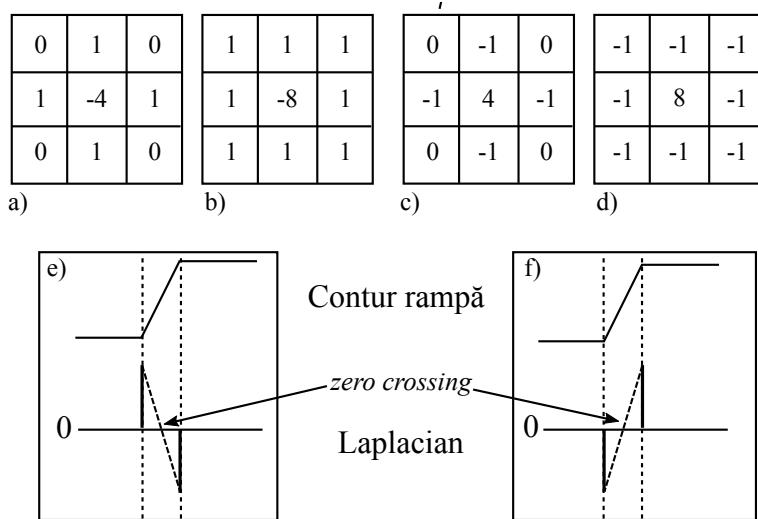


Figura 9: a) Filtrul Laplace b) Filtrul Laplace extins și la vecinii diagonali. c) Filtrul Laplace de la a) cu centrul pozitiv. d) Filtrul Laplace de la b) cu centrul pozitiv. e) Contur rampă cu rezultatul produs de Laplacian considerând măștile a) și b). f) Contur rampă cu rezultatul produs de Laplacian considerând măștile c) și d)

*Observații:*

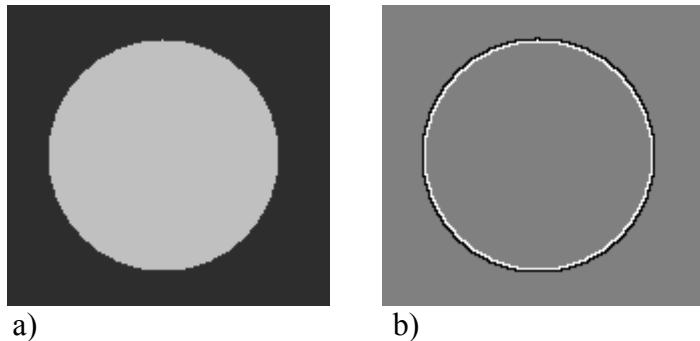


Figura 10: a) Imagine de test reprezentând un disc de culoare deschisă pe fundal închis. b) Imaginea filtrată cu filtrul Laplace din fig.9 d) în care nivelele de gri au fost translatație cu 128.

1. Derivata de ordinul 2 produce margini duble și unele unele la tranziția de la fond la contur - intrarea în rampă = valoare pozitivă - și una la tranziția de la contur la obiect - ieșirea din rampă = valoare negativă. Pentru valorile aflate în rampă, derivata a două este 0. Acest rezultat este semnificativ din următorul punct de vedere. Considerând cele două valori ale derivatei a două la intrare în rampă-pozitivă - și respectiv ieșirea din rampă - negativă, intersecția segmentului care le unește cu axa  $0x$  se numește *zero-crossing* și poate fi utilizată pentru determinarea centrului unui contur gros și astfel poate fi considerată ca fiind de fapt punct de contur, în obținerea unui contur subțire - fig. 9 e).
2. Rezultatele obținute prin aplicarea unui operator de derivare sunt influențate de existența zgomotului, de aceea, în cazul imaginilor afectate de zgomot este necesară aplicarea în prealabil a unui filtru trece-jos pentru reducerea acestuia.
3. Variante ale măștilor de filtrare sunt cele din fig. 9 c) și d), în care elementul central are semnul pozitiv. În această situație valoarea produsă la intrarea în rampă este negativă, iar cea produsă la ieșirea din rampă este pozitivă - fig. 9 f).

În figura 10 este reprezentată filtrarea unei imagini de test cu filtrul Laplace din fig.9 d). Valorile obținute au fost apoi translatație cu 128 pentru a permite și reprezentarea valorilor negative. Se observă producerea de contururi duble: pe fiecare contur se află tonuri de gri închise, reprezentând valorile negative, deci intrarea în rampă precum și tonuri de gri deschise, reprezentând valorile pozitive, deci ieșirea din rampă. Valorile de gri apropiate de 128 reprezintă zonele uniforme din imagine, în care Laplacianul este 0.

## 4.2 Operatorul *Marr-Hildreth*

Luând în considerare aspectele teoretice prezentate mai sus, detectarea contururilor bazată pe operatorul Laplace poate fi realizată în următoarele etape:

1. Netezire cu un filtru trece-jos, pentru a reduce zgomotul, deoarece filtrul Laplace este

foarte sensibil la zgomot. De obicei se folosește pentru aceasta un filtru Gaussian.

2. Aplicarea operatorului Laplace pentru a obține informația de contur. Trebuie avut în vedere faptul că operatorul Laplace produce atât valori negative cât și valori pozitive, ambele tipuri de valori fiind necesare la detectarea marginilor.
3. Detectare efectivă folosind un algoritm de *zero-crossings*. Un astfel de algoritm este descris în cele ce urmează.

### ***Zero-crossings***

Un algoritm pentru determinarea contururilor folosind ***zero-crossings*** este acela bazat pe un set de 8 măști  $3 \times 3$ :

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	A <sub>1</sub>		B <sub>1</sub>		A <sub>1</sub>	
	(x,y)			(x,y)		A <sub>2</sub>	(x,y)	B <sub>2</sub>	B <sub>2</sub>	(x,y)	A <sub>2</sub>
B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>3</sub>		B <sub>3</sub>		A <sub>3</sub>	
A <sub>2</sub>	A <sub>3</sub>		B <sub>2</sub>	B <sub>3</sub>			A <sub>1</sub>	A <sub>2</sub>		B <sub>1</sub>	B <sub>2</sub>
A <sub>1</sub>	(x,y)	B <sub>3</sub>	B <sub>1</sub>	(x,y)	A <sub>3</sub>	B <sub>1</sub>	(x,y)	A <sub>3</sub>	A <sub>1</sub>	(x,y)	B <sub>3</sub>
	B <sub>1</sub>	B <sub>2</sub>			A <sub>1</sub>	A <sub>2</sub>			A <sub>2</sub>	A <sub>3</sub>	

Pentru fiecare pixel  $(x, y)$  se consideră pe rând fiecare mască centrată în  $(x, y)$ . Dacă pentru una dintre măști ( $A_1 \geq t$  și  $B_1 \leq -t$ ) sau ( $A_2 \geq t$  și  $B_2 \leq -t$ ) sau ( $A_3 \geq t$  și  $B_3 \leq -t$ ), atunci se consideră că  $(x, y)$  este punct de contur și se colorează cu alb în imaginea rezultat. Dacă nu, se consideră că  $(x, y)$  este fundal și se colorează cu negru.  $t$  este un prag ales astfel, încât să se reducă erorile datorate zgomotului.

Operațiile de la pasul (1) și de la pasul (2), netezirea cu un filtru Gaussian și aplicarea filtrului Laplace pot fi combinate într-un singur pas, ceea ce duce la eficientizarea algoritmului. Acest lucru se realizează construind o mască pe baza operatorului *Marr-Hildreth*, numit și *LoG - Laplacian of Gaussian*  $\nabla^2 G(x, y)$ .

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

unde  $G$  este funcția Gaussiană

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

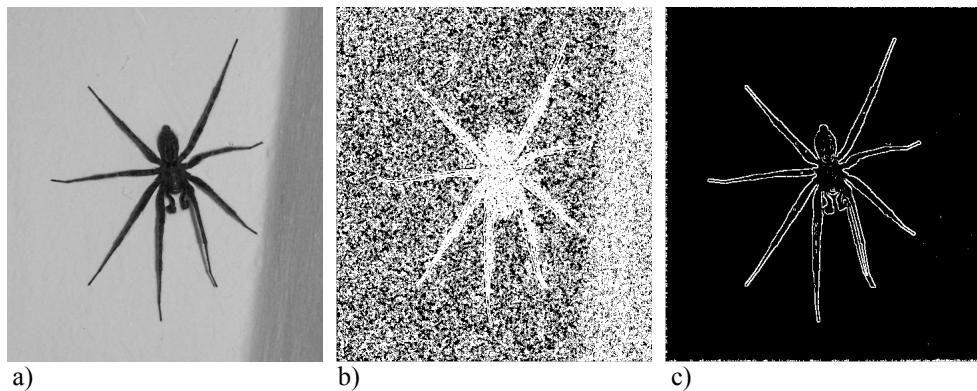
În urma derivării se obține operatorul LoG:

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

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



**Exemplu** de aplicare a operatorului Laplace fără netezire prealabilă, precum și după netezirea imaginii cu filtrul Gauss.



### 4.3 Evidențierea contururilor - *Unsharp mask*

Din secțiunea anterioară se observă faptul că filtrul Laplace produce în proximitatea contururilor câte o valoare pozitivă și una negativă. Astfel, pe baza acestui filtru pot fi scoase în evidență contururile și anume utilizând operatorul numit *unsharp mask* dat de:

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

Operatorul de *unsharp mask* determină ca la intrarea în rampă valorile închise aflate pe marginea conturului să devină mai închise iar la ieșirea din rampă valorile deschise de pe marginea conturului să devină mai deschise și astfel crește contrastul în apropierea conturului.

Acest operator poate fi sintetizat printr-o mască de filtrare, permitând astfel un calcul mai rapid.

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

Există variante de măști cu valori mai mari în centru, de exemplu 5, 7 sau 9. Acestea accentuează efectul.



**Exemplu** de aplicare a operatorului *unsharp mask* asupra unei imagini în tonuri de gri.



a) Imaginea originală. Imaginea după aplicarea operatorului de *unsharp mask* b) cu filtrul Laplace din fig.9 c), c) cu filtrul Laplace din fig.9 d). Se observă că masca în care se consideră și operatorii diagonali produce rezultate cu contrast mai mare.



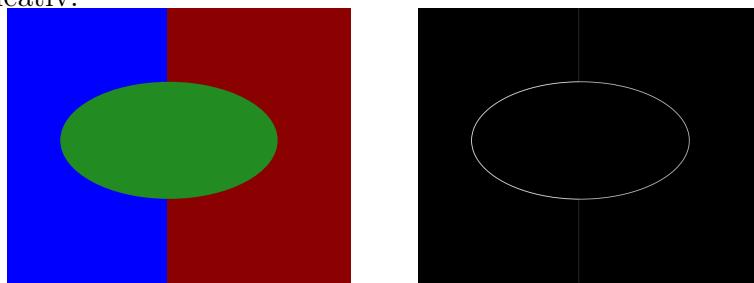
**To Do:** Implementați operatorul UnsharpMask folosind un filtru Laplace în *framework*-ul de laborator.

## 5 Filtrarea trece sus a imaginilor color

Cea mai simplă variantă a tratării imaginilor color ar fi aceea de a considera transforma imaginea într-una în tonuri de gri și de a aplica un algoritm de detectare a marginilor descris mai sus, de exemplu pe baza gradientului și a filtrului Sobel. Rezultatele astfel obținute pot fi problematice, mai ales acolo unde există culori similare din punct de vedere al intensității culorii. În plus, două suprafete vecine de culori complet diferite pot avea niveluri de gri foarte similare, algoritmul ne detectând acolo margini. Este deci evident că pentru o detectie cât mai bună culoarea este un descriptor important.



**Exemplu** de imagine sintetică RGB, în care prin transformarea în tonuri de gri și aplicarea filtrului Sobel, marginea dintre două suprafete colorate diferit, dar cu intensitate similară nu este detectată, gradientul fiind nesemnificativ.



O variantă mai bună din punct de vedere al rezultatului, propusă în ??, este acela de a considera toate cele trei canalele de culoare separat și de a calcula gradientul pentru fiecare în parte folosind, de exemplu, Sobel:

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

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

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

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*).

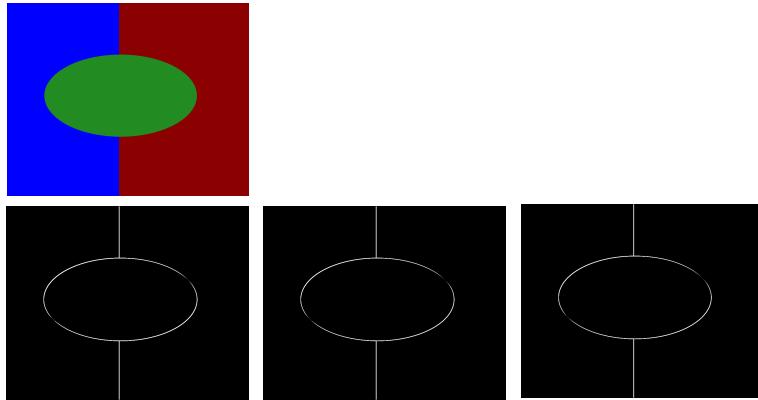
Pe baza valorilor pe cele trei canale de culoare,  $E_R$ ,  $E_G$  și  $E_B$  se poate considera vectorul *Edge*:

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

care combină *tăriile* marginilor pentru fiecare canal de culoare. Interpretarea acestui vector pentru a obține imaginea rezultată poate fi făcută în mai multe moduri. De exemplu se poate obține o imagine RGB, în care fiecărui pixel  $(x, y)$  i se asociază culoarea data de  $Edge(x, y)$ . În acest mod se realizează obținerea marginilor cu instrumentul corespunzător din Photoshop. Din punct de vedere practic, acest lucru nu este însă foarte relevant.



**Exemplul 1.** Imagine sintetică RGB în care marginile au fost detectate folosind gradientii combinați prin normele  $L_1$  (stânga),  $L_1$  (centru) și  $L_{infty}$  (dreapta).



Pentru a obține rezultate similare celor din domeniul imaginilor în tonuri de gri, când detectarea de margini conduce la obținerea unei imagini de gradient și în final, prin prăguire, a unei imagini binare, ce conține doar marginile relevante, ar trebui asociat fiecărui pixel  $(x, y)$  o singură valoare, care estimează *tăria* marginii color în punctul respectiv. Acest lucru poate fi făcut considerând o normă a vectorului  $Edge(x, y)$ . Variantele de luat în considerare ar fi:

$$(a) E_2(x, y) = L_2(Edge(x, y)) = \|Edge(x, y)\|_2 = \sqrt{E_R(x, y)^2 + E_G(x, y)^2 + E_B(x, y)^2}$$

$$(b) E_1(x, y) = L_1(Edge(x, y)) = ||Edge(x, y)||_1 = |E_R(x, y)| + |E_G(x, y)| + |E_B(x, y)|$$

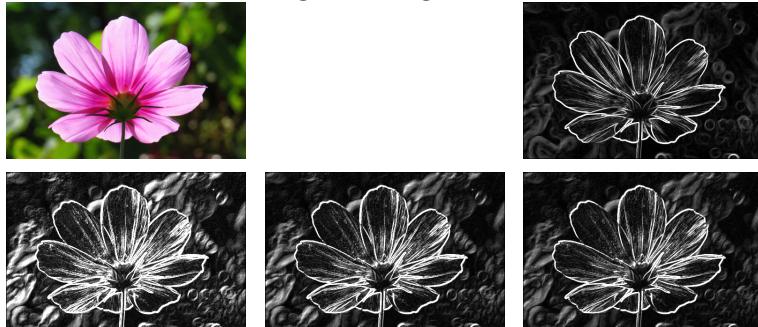
$$(c) E_\infty(x, y) = L_{infty}((Edge(x, y))) = ||Edge(x, y)||_\infty = \max(|E_R(x, y)|, |E_G(x, y)|, |E_B(x, y)|)$$



**To Do:** Implementați în *framework*-ul de la laborator o funcție pentru detectarea de marginilor folosind gradienții pe cele 3 canale de culoare, prin utilizarea uneia dintre normele propuse mai sus pentru estimarea cărării marginii.



**Exemplul 2.** Imagine naturală RGB în care marginile au fost detectate folosind gradienții combinați prin normele  $L_1$  (stânga),  $L_1$  (centru) și  $L_{infty}$  (dreapta). Comparativ în colțul dreapta sus este rezultatul aplicării filtrului Sobel pe varianta în tonuri de gri a imaginii color.



Se observă din exemple că, rezultatele folosind diferitele norme sunt relativ asemănătoare. Dat fiind că norma  $L_{infty}$  este cea mai puțin costisitoare, poate fi folosită cu succes.

Problema principală a acestei abordări este determinarea direcției gradientului., dat fiind că, acest lucru nu poate fi făcut decât pe fiecare canal de culoare îndeplinește. O variantă de aproximare propusă în ?? ar putea fi aceea de a considera direcția pe canalul de culoare, pentru care norma gradientului este cea mai mare.

## 5.1 Gradient multidimensional

O margină, așa cum s-a specificat la începutul capitolului, reprezintă o variație locală semnificativă a culorii. În cazul imaginilor în tonuri de gri, acest lucru era ușor de determinat, datorită faptului că o culoare este reprezentată de o valoare scalară - nivelul de gri. În acest caz gradientul poate fi folosit pentru determinarea variației locale, atât ca valoare (norma gradientului) cât și ca direcție. În cazul imaginilor color fiecare culoare este un triplet  $(r, g, b)$ . S-a discutat în secțiunea premergătoare, cu poate fi estimată variația folosind ori operatorul Sobel separat pe cele trei canale, ori variații ale operatorului Sobel utilizând distanța euclidiană sau unghiul vectorial. Niciuna dintre aceste variante nu îmi permite însă estimarea direcției conturului, informație esențială sper exemplu pentru operatorul Canny.

O metodă mai bună de estimare a variației locale în cazul imaginilor color și care permite

și estimarea direcției de este aceea trata o imagine color ca pe un *câmp vectorial* [1]. Adică o imagine color  $f(x, y) = (R(x, y), G(x, y), B(x, y))$  este o funcție de două variabile  $(x, y)$  ale cărei valori sunt vectori tridimensionali. Pentru o astfel de imagine putem calcula derivatele parțiale:

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

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

Pe baza acestora se construiește matricea Jacobiană dată de:

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

unde

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

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

Din aceste formule se poate calcula variația culorii după orice direcție dată print-un unghi  $\theta$  față de axa orizontală prin înmulțirea Jacobianului cu vectorul  $(\cos \theta, \sin \theta)^T$ , reprezentând *gradientul direcțional* [1]:

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

Se demonstrează în [4] că, unghiul  $\theta$  pentru care norma gradientului direcțional  $\text{grad}_\theta(x, y)$  este maximă, este dat de direcția vectorului propriu corespunzător valorii proprii celei mai mari a matricii  $M = J_f^T J_f$ :

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

unde

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

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

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

Valorile pentru  $\partial R(x, y)/\partial x$ ,  $\partial G(x, y)/\partial x$  și  $\partial B(x, y)/\partial x$  pot fi estimate folosind operatorul Sobel obișnuit pe fiecare canal de culoare împărțit.

**Calculul vectorilor și valorilor proprii** pentru matricea  $M$  se realizează prin:

valori proprii: - soluțiile  $\lambda_1, \lambda_2$  ale ecuației

$$|(M - \lambda I)| = 0 \quad (1)$$

$I$  - matricea identică.

vectorii proprii corespunzători  $V_1, V_2$ :

$$MV_1 = \lambda_1 V_1$$

$$MV_2 = \lambda_2 V_2$$

Înlocuind în 1  $M$  cu matricea efectivă și calculând determinantul obținem:

**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:

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

Vector propriu asociat valorii  $\lambda_1$ :

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

**Orientarea vectorului  $v_1$  [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 etape de calcul:**

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

- Dacă este necesar (ex. Canny) se calculează direcția de variație maximă (formulă)
- Dacă  $\text{grad}(x, y) > T$ ,  $T$  - threshold, atunci se consideră  $(x, y)$  pixel de contur, altfel se consideră pixel de fundal.



**Exemplul 1** pentru determinarea imaginii gradient cu algoritmul prezentat. În figură, în dreapta este o imagine RGB, în centru este rezultatul obținut cu operatorul Sobel modificat (distanță Euclidiană), iar în dreapta este rezultatul obținut prin determinarea direcției de variație maximă a culorii.



**Exemplul 2** pentru determinarea imaginii gradient cu algoritmul prezentat. În figură, în dreapta este o imagine RGB, în centru este rezultatul obținut cu operatorul Sobel modificat (distanță Euclidiană), iar în dreapta este rezultatul obținut prin determinarea direcției de variație maximă a culorii.



## 5.2 Operatorul Canny pentru imagini color



**Exemplu.** Imagine color împreună cu rezultatul aplicării operatorului Canny descris.



Pentru a obține margini de grosime 1 pixel similar ca în cazul imaginilor în tonuri de gri, algoritmul Canny descris anterior poate fi adaptat pentru imagini color folosind direcția de variație maximă a culorii în modul descris mai jos.

## **Etape Canny pentru imagini color**

- Reducerea zgromotului - filtru Gauss pe fiecare canal ( $\sigma = 1$  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*



### **To Do:**

- Implementați în *framework*-ul de la laborator o funcție pentru detectarea de marginilor folosind Jacobianul, aşa cum a fost descris în curs.
- Căutați în documentație funcția OpenCV corespunzătoare pentru operatorul Canny.



## Să ne reamintim

- Un filtru trece-sus reduce frecvențele joase ( prezente în zone uniforme) și păstrează frecvențele înalte (contururi / zgomot)
- Pentru construirea de filtre trece-sus se poate folosi gradientul (derivate parțiale de ordinul I), derivatele parțiale fiind approximate prin diferențe finite. Astfel de filtre apar în perechi, un filtru pentru derivata în direcția orizontală și unul în direcția verticală. Cel mai cunoscut filtru bazat pe gradient este filtrul Sobel.
- Filtrarea cu un filtru trece-sus de tip gradient oferă informații asupra marginilor (*edges*) prezente în imagini. Pentru a obține o imagine binară, care conține doar margini trebuie folosită o operație de praguire (*thresholding*) asupra imaginii de gradient.
- Operatorul Canny care are la bază detectarea de margini cu filtrul Sobel permite obținerea de margini de grosime 1 pixel, cu fragmentare redusă.
- Filtrarea trece sus se poate realiza și cu ajutorul operatorului Laplace, bazat pe derivata de ordin 2. În acest caz, pentru detectarea efectivă a marginilor se utilizează un algoritm numit *zero-crossing*.
- Filtrul Laplace este foarte sensibil la zgomot, de aceea se combină cu filtrul Gauss, obținându-se operatorul *Marr-Hildreth*.
- Filtrarea trece-sus pentru imagini color se poate realiza cel mai eficient folosind gradientul multidimensional și detectarea direcției de variație maximă a culorii, cu ajutorul vectorilor și valorilor proprii ale matricii obținute pe baza Jacobianului (vezi descriere)



## Rezumat: în acest capitol

- a fost prezentată filtrarea în domeniul spațial, bazată pe operația de convoluție
- a fost discutată filtrarea trece-jos, care are ca scop reducerea zgomotului în imagini.
- au fost prezentate cele mai cunoscute tipuri de filtre trece-jos în domeniul spațial (filtrul medie aritmetică, filtrul Gaussian, filtrul median)
- au fost prezentate filtre avansate trece-jos, dintre care cel mai important este filtrul bilateral
- a fost discutată filtrarea trece-sus pentru detectarea de margini în imagini
- au fost prezentate filtre bazate pe gradient (Prewitt, Sobel), împreună cu operatorul Canny, precum și filtre bazate pe Laplacian (Marr-Hildreth)
- s-a discutat problema filtrării pentru imagini color



**Test de autoevaluare:**

## Bibliografie

- [1] Burger W, Burger M.J „Principals of Digital Image Processing. Core Algorithms”, Springer 2009
- [2] Burger W, Burger M.J „Principals of Digital Image Processing. Advanced Methods”, Springer 2013
- [3] Gonzalez R C; Woods R.E, „Digital Image Processing, global edition”, Perason 2018
- [4] Cumani
- [5] D. Richter „Digital Image Processing” 2019, curs pentru studenti.