

CUPRINS

Contents

| | |
|--|----|
| Detectia frontierelor din imagini (Edge detection) | 1 |
| 1. Detectia pixelilor de front bazata pe Gradient-ul imaginii..... | 2 |
| 1.1. Detectorul Roberts (cruce)..... | 5 |
| 1.2. Detectorul Sobel..... | 6 |
| 1.3. Detectorul Prewitt | 7 |
| 1.4. Detectia pixelilor de front în imagini color..... | 8 |
| 2. Detectia pixelilor de front bazata pe Laplacian-ul imaginii | 10 |
| 2.1. Laplacian-ul unei imagini | 10 |
| 2.2. Operatorul LoG (Laplacian of Gaussian) | 12 |
| 3. Extragerea frontierelor | 16 |
| 3.1. Binarizarea matricei amplitudinilor | 16 |
| 3.2. Detectorul Canny | 17 |
| 3.3. Metoda Roberts..... | 21 |
| 3.4. Extragerea frontierelor bazata pe predictia punctului urmator | 22 |
| 3.5. Transformata Hough (detectia liniilor drepte) | 25 |

Detectia frontierelor din imagini (Edge detection)

Frontierele sunt zone din imagine caracterizate prin variatii bruste ale intensitatii. Ele pot reprezenta obiecte subtiri (linii) sau contururi ale unor zone din imagine cu caracteristici de intensitate similare. Detectia frontierelor este o problema fundamentala in prelucrarea imaginilor, deoarece permite extragerea informatiei utile dintr-o imagine, necesara in analiza imaginilor.

Majoritatea metodelor de detectie a frontierelor presupun efectuarea urmatoarelor etape:

- 1) Detectia pixelilor de front.

Un **pixel de front** (pe scurt, **front**) este un pixel in care intensitatea se schimba brusc. Din pacate, nu orice pixel de front este pixel de frontiera. El poate fi un punct de zgomot. Din acest motiv, unele detectoarele de pixeli de front opereaza asupra unei imagini netezite. Dar, filtrele de netezire diminueaza nu numai intensitatea punctelor de zgomot ci si pe aceea a pixelilor de frontiera, ingreunand detectia punctelor de front.

- 2) Se elimina dintre pixelii de front aceia care, pe baza unor criterii, nu sunt pixeli de frontiera.
- 3) Se conecteaza pixelii de frontiera pentru a forma contururi (frontiere).

Pentru conectarea pixelilor de frontiera se folosesc 2 tipuri de metode:

- *Metode locale*, care folosesc relatiile (bazate pe intensitate ale) fiecarui pixel din imagine cu pixelii vecini. Frontiera este contruita iterativ, intr-un proces de "urmarire" a punctelor de frontiera.
- *Metode globale*, care utilizează informații globale, cum ar fi cunoasterea formei geometrice a frontierelor sau a ecuatiei matematice (ex. Transformata Hough).

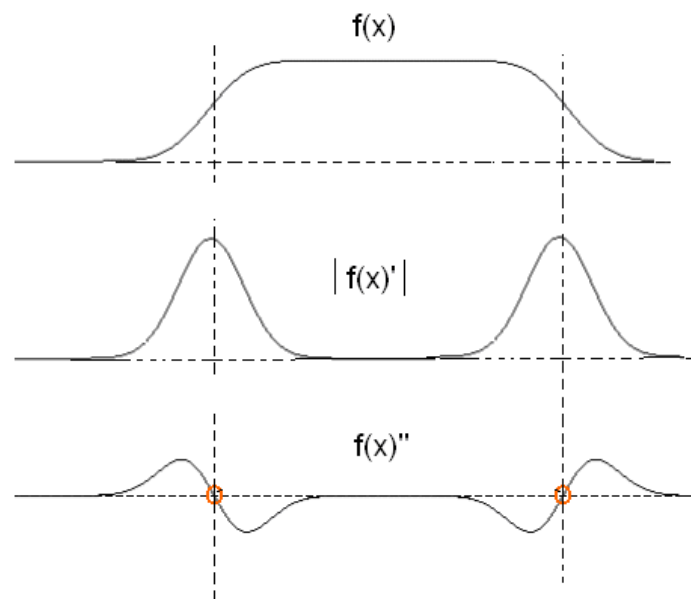
Majoritatea detectoarelor de frontiere utilizeaza derivata I-a si derivata a 2-a a functiei imagine, stiind ca intr-un punct de front (vezi figura urmatoare):

- Derivata I-a are un maxim sau un minim local
- Derivata a 2-a trece prin zero.

În cazul 2-D, pentru o funcție $f(x,y)$, prima derivată corespunde Gradientului, iar a 2-a derivată corespunde Laplacianului.

Deteția de fronturi în imagini se bazează pe operatori care aproximează în planul discret Gradientul sau Laplacianul imaginii.

- Detecțiile bazate pe gradient produc matricea amplitudinii gradientului în fiecare punct al imaginii de intrare. De asemenea, ele pot produce și matricea direcției gradientului în fiecare punct al imaginii de intrare.
- Detecțiile bazate pe Laplacian sunt *izotropice* – rezultatul lor este același indiferent de direcțiile fronturilor din imagine. Un detector care folosește Laplacianul produce o matrice în care punctele de front sunt puncte de tranziție prin valoarea zero.



Profilul 1D al unui front

1. Detectia pixelilor de front bazata pe Gradient-ul imaginii

Vectorul gradient reprezinta directia si marimea variatiei maxime de intensitate intr-un punct al unei imagini. El este definit în funcție de derivatele parțiale ale funcției imagine, $f(x,y)$:

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

Notam cu $D_x = \frac{\partial f(x, y)}{\partial x}$ și $D_y = \frac{\partial f(x, y)}{\partial y}$, atunci $\begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} D_x \\ D_y \end{bmatrix}$

Amplitudinea gradientului în pixelul (x,y) este:

$$D(x, y) = [D_x^2(x, y) + D_y^2(x, y)]^{1/2}$$

Ea reprezinta **puterea frontului** (marimea variatiei de intensitate) în pixelul respectiv si este adesea **aproximata prin**:

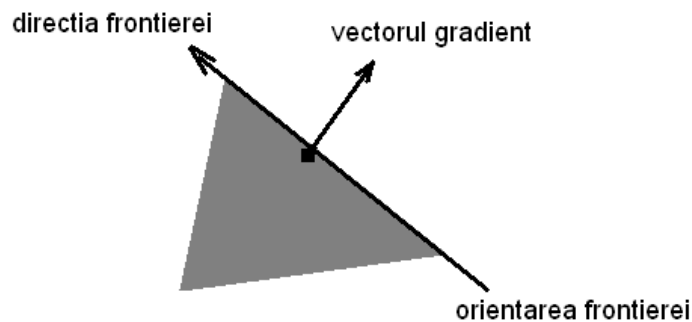
$$D(x, y) = |D_x| + |D_y|$$

$$D(x, y) = \text{Max}(|D_x|, |D_y|)$$

Direcția gradientului (frontului) este:

$$\theta = \text{arctg}\left(\frac{D_y}{D_x}\right), \text{ pentru } D_x \neq 0$$

Vectorul gradient este perpendicular pe direcția frontierei:



Punctele de front dintr-o imagine sunt detectate calculând gradientul în fiecare pixel și identificând acei pixeli pentru care amplitudinea gradientului este mai mare decât un prag dat.

Pentru **implementarea discretă a gradientului** au fost propuși mai mulți operatori. De exemplu:

Aproximarea prin *diferențe finite*. Cele 2 derivate se aproximează prin:

$$D_{1x} = \frac{\partial f(x, y)}{\partial x} \approx \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

$$D_{1y} = \frac{\partial f(x, y)}{\partial y} \approx \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$

Aceste aproximări corespund *corelării* funcției imagine cu măști de convoluție de forma:

$$[-1 \quad 1] \text{ și } \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

O altă aproximare este:

$$D_{2x} \approx \frac{f(x + \Delta x, y) - f(x - \Delta x, y)}{2(\Delta x)}$$

$$D_{2y} \approx \frac{f(x, y + \Delta y) - f(x, y - \Delta y)}{2(\Delta y)}$$

D_{2x} și D_{2y} sunt aproximări mai bune în mijlocul intervalului, adică în (x, y) și corespund corelării funcției imagine cu măștile:

$$\frac{1}{2}[-1 \ 0 \ 1] \text{ și } \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Cei mai utilizați detectori de fronturi bazati pe gradient sunt: Roberts (numit și operatorul *cruce*), Sobel și Prewitt. Ei sunt definiți pentru imagini în mai multe nivele de gri, dar utilizarea lor poate fi extinsă pentru imagini color.

1.1. Detectorul Roberts (cruce)

Componentele gradientului intr-un pixel (x,y) sunt approximate prin:

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

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

D+ si D- se calculeaza cu urmatoarele 2 masti de convolutie:

| | |
|-----|----|
| + 1 | 0 |
| 0 | -1 |

| | |
|----|----|
| 0 | +1 |
| -1 | 0 |

D- D +

Cele 2 masti sunt proiectate pentru detectia maxima a fronturilor avand directii de 45 si 135 grade (directii perpendiculare, de unde si numele de "operatorul cruce"):

Amplitudinea gradientului in (x,y) este aproximata prin:

$$D(x,y) = |D_-(x,y)| + |D_+(x,y)|$$

Orientarea frontului in pixelul (x,y) este data de:

$$\theta = \arctg(D_+ / D_-) - 3\pi/4$$

Orientarea (unghiul) zero inseamna ca directia contrastului maxim de la negru la alb este de la stanga la dreapta in imagine, celelalte unghiuri fiind masurate in sensul acelor de ceas fata de unghiul zero.

De regula, iesirea produsa de acest detector este numai matricea amplitudinilor, vizualizata sub forma unei imagini in nivele de gri. In acest caz, cele doua componente ale gradientului pot fi calculate si adunate intr-un singur pas, utilizand urmatoarea masca de pseudo-convolutie:

| | |
|-----|----|
| + 1 | +1 |
| -1 | -1 |

Principalul avantaj al utilizării acestui detector este simplitatea calculelor: operații de adunare și scădere cu valorile a numai 4 pixeli.

Dezavantajul: deoarece utilizează un nucleu foarte mic, este foarte sensibil la zgomot. De asemenea, răspunsul său la frontiere reale este slab dacă acestea nu sunt foarte “abrupte” (o tranziție de intensitate foarte mare).

1.2. Detectorul Sobel

Gradientul este aproximat prin convoluția imaginii cu următoarele masti:

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

Dx

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

Dy

Mastile sunt proiectate pentru răspuns maxim la fronturi cu direcțiile verticale și orizontale. Ele pot fi aplicate separat imaginii de intrare, fiind apoi folosite pentru calculul amplitudinii și al direcției frontului în fiecare pixel al imaginii de intrare.

Amplitudinea este aproximată prin:

$$D(x,y) = |Dx| + |Dy|$$

iar direcția prin:

$$\theta = \arctg(Dy / Dx), \text{ pentru } Dx \neq 0$$

$$\pm 90, \text{ pentru } Dx = 0, Dy \neq 0$$

$$0, \text{ pentru } Dy = 0$$

Unghiul zero inseamna ca directia contrastului maxim de la negru la alb este de la stanga la dreapta in imagine, celelalte unghiuri fiind masurate in sens trigonometric (invers acelor de ceas) fata de unghiul zero.

Atunci cand iesirea detectorului de fronturi este numai matricea amplitudinilor, componentele Dx si Dy pot fi calculate si adunate intr-un singur pas, folosind operatorul de pseudo-convolutie:

$$D(x,y) = | (P_3 + 2xP_6 + P_9) - (P_1 + 2x P_4 + P_7) | + | (P_1 + 2xP_2 + P_3) - (P_7 + 2x P_8 + P_9) |$$

unde P₁,..., P₉ sunt pixelii acoperiti de masca de convolutie, ca in figura urmatoare:

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

Operatorul Sobel solicita mai multe calcule decat operatorul Roberts, dar masca sa de convolutie fiind mai mare, netezeste mai mult imaginea si de aceea este mai putin sensibil la zgomot. Produce valori de amplitudine mai mari decat cele produse de operatorul Roberts, pentru aceleasi frontiere.

Matricea amplitudinilor produsa de acest operator poate fi vizualizata ca o imagine in nivele de gri.

Datorita efectului de netezire pe care il are, frontierele apar adesea in aceasta imagine ca linii a caror latime este de mai multi pixeli. Pentru subtierea frontierei este necesare o postprocesare, cum ar fi aceea de "histerezis", din algoritmul Canny.

1.3. Detectorul Prewitt

Este asemanator detectorului Sobel. Foloseste mastile:

| | | |
|----|---|----|
| -1 | 0 | +1 |
| -1 | 0 | +1 |
| -1 | 0 | +1 |

Dx

| | | |
|----|----|----|
| +1 | +1 | +1 |
| 0 | 0 | 0 |
| -1 | -1 | -1 |

Dy

1.4. Detectia pixelilor de front in imagini color

Se poate baza pe diferentele in stralucire sau in nuanza (cele 3 proprietati ale luminii (HSV) : nuanza, saturatie, stralucire/intensitate).

- Detectia bazata pe stralucire se trateaza la fel ca pentru imagini cu mai multe nivele de gri.
- Detectia bazata pe nuanze, identifica punctele de front pe baza diferentelor cromatice.

In cazul reprezentarii culorilor in sistemul (R,G,B), se pot detecta fronturile aplicand un operator de detectie separat pe cele trei componente si apoi insumand rezultatele, dar aceasta metoda poate conduce la erori. Amplitudinea se aproximeaza prin:

$$D = |D_R| + |D_G| + |D_B|$$

O imagine in nivele de gri este reprezentata printr-o functie scalara, $f(x,y)$. Gradientul sau intr-un punct (x,y) este un vector a carui directie indica rata maxima de schimbare a functiei f in punctul (x,y) .

Culoarea este un vector intr-un spatiu tri-dimensional. O definitie mai exacta a gradientului unei imagini color trebuie sa se bazeze pe gradientul (marime si directie) vectorului culoare.

- Fie r, g, b versorii axelor R,G,B din spatiul RGB al culorilor.
- Fie vectorii u si v care reprezinta variatia de culoare RGB intr-un pixel (x,y),

$$u = \frac{\partial R}{\partial x} r + \frac{\partial G}{\partial x} g + \frac{\partial B}{\partial x} b$$

$$v = \frac{\partial R}{\partial y} r + \frac{\partial G}{\partial y} g + \frac{\partial B}{\partial y} b$$

si produsele scalare:

$$g_{xx} = u * u = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2$$

$$g_{yy} = v * v = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2$$

$$g_{xy} = u * v = \frac{\partial R}{\partial x} * \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} * \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} * \frac{\partial B}{\partial y}$$

Se poate arata ca directia de schimbare maxima a culorii in punctul (x,y) este data de unghiul

$$o = \frac{1}{2} \arctg \frac{2g_{xy}}{g_{xx} - g_{yy}}$$

iar $\sqrt{F(o)}$ este puterea frontului (rata maximă de schimbare), unde

$$F(o) = \frac{1}{2} \left[(g_{xx} + g_{yy}) + \cos(2o)(g_{xx} - g_{yy}) + 2g_{xy} * \sin(2o) \right]$$

Prin aplicarea unui prag peste $\sqrt{F(o)}$ se obține harta frontului.

Derivatele partiale ale functiilor R(x,y), G(x,y) si B(x,y) se aproximeaza prin:

$$\frac{\partial f_i}{\partial x} \approx \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * f_i$$

$$\frac{\partial f_i}{\partial y} \approx \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * f_i$$

unde “*” desemnează convoluția discretă a funcției imagine, iar f_i este $R(x,y)$, $G(x,y)$, $B(x,y)$.

2. Detectia pixelilor de front bazata pe Laplacian-ul imaginii

2.1. Laplacian-ul unei imagini

Operatorii pentru determinarea gradientului presupun calculul componentelor D_x și D_y apoi combinarea lor pentru a obține puterea (amplitudinea) frontului.

Operatorul Laplacian, care detectează tranzițiile bruste de intensitate din imagine, cere mai puțin calcule. El este definit în funcție de derivatele parțiale de ordinul 2, astfel :

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

Derivatele parțiale de ordinul al 2-lea, de-a lungul direcțiilor x și y , sunt approximate prin diferențe finite astfel:

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

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

Pentru $\Delta x = \Delta y = 1$. Rezultă :

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

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

$$\nabla^2 f = f(x+1, y) - 4f(x, y) + f(x-1, y) + f(x, y+1) + f(x, y-1).$$

Laplacian-ul poate fi implementat prin urmatoarea masca de convolutie:

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

Aplicarea operatorului consta in convolutia imaginii cu masca respectiva. Intrarea operatorului este o imagine, $f(x,y)$, in mai multe nivele de gri iar iesirea, $g(x,y)$, de asemenea, o imagine in mai multe nivele de gri:

$g(x,y) = 0$ in zonele de intensitate constanta

<0 sau > 0 in vecinatatea frontierei

Laplacian-ul unei imagini este utilizat si pentru implementarea unor filtre de imbunatatire a imaginilor prin evidentierea frontierelor (image sharpening). Astfel sunt filtrele cu mastile de convolutie:

| | | | | | | | | |
|----|----|----|----|----|----|---|----|---|
| 0 | -1 | 0 | -1 | -1 | -1 | 1 | 1 | 1 |
| -1 | 4 | -1 | -1 | 8 | -1 | 1 | -8 | 1 |
| 0 | -1 | 0 | -1 | -1 | -1 | 1 | 1 | 1 |

Alte filtre de imbunatatire a imaginilor bazate pe Laplacian sunt definite astfel:

$$g(x,y) = \begin{matrix} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{matrix}$$

Adunand sau scazand imaginea rezultata prin aplicarea Laplacianului se obtine o imagine cu frontierele evidentiate.

Pentru detectia frontierelor folosind Laplacianul, se poate proceda in mai multe moduri:

1) Se aplica un prag peste $g(x,y)$, rezultand o imagine binara, $B(x,y)$:

$$B(x,y) = \begin{matrix} 1, \text{ daca } |g(x,y)| > \text{prag} \\ 0, \text{ altfel} \end{matrix}$$

Cat de aproape de frontiera reala sunt pixelii cu valoarea 1 in $B(x,y)$ depinde de alegerea pragului. In functie de pragul ales, frontiera va fi localizata fie in partea luminata fie in cea intunecata a frontierei. O abordare mai exacta consta in examinarea pixelilor de ambele parti ale granitei si alegerea acelor care au cea mai mica valoare absoluta a Laplacianului, considerand ca aceste puncte sunt cele mai apropiate de punctele de trecere prin zero.

2) Sunt considerate puncte de frontiera acele puncte in care $g(x,y) \neq 0$ si gradientul are valoarea locala maxima (sau peste un prag dat). Directia frontului se obtine utilizand directia in care are loc trecerea prin zero a Laplacianului.

2.2. Operatorul LoG (Laplacian of Gaussian)

Mai este cunoscut sub numele: "Marr edge detector" si "Zero Crossing detector".

Operatorul Laplacian este foarte sensibil la zgomot. Pentru reducerea acestui efect, inainte de aplicarea operatorului Laplacian se poate aplica asupra imaginii un filtru de netezire pentru reducerea componentelor de inalta frecventa din imagine.

Operatorul LoG imbină următoarele trei operații:

- Filtrarea, prin aplicarea unui filtru Gaussian
- Îmbunătățirea, prin aplicarea operatorului Laplacian
- Detectia fronturilor, ca puncte de trecere prin zero a Laplacianului

Fie $g(x,y)$ filtrul Gaussian:

$$g(x,y) = (1 / (2\pi\sigma^2)) \exp(-(\mathbf{x}^2 + \mathbf{y}^2) / (2 \sigma^2))$$

Aplicarea sa asupra imaginii constă în convoluția funcției $g(x,y)$ cu funcția imagine, $f(x,y)$.

Convoluția discretă a celor două funcții este definită astfel:

$$g(x,y) * f(x,y) = \sum_{i=-a}^a \sum_{j=-b}^b w(i,j) * f(x+i, y+j)$$

unde w este masca de convoluție care aproximează funcția $g(x,y)$ în planul discret.

Laplacianul convoluției celor două funcții se poate scrie astfel:

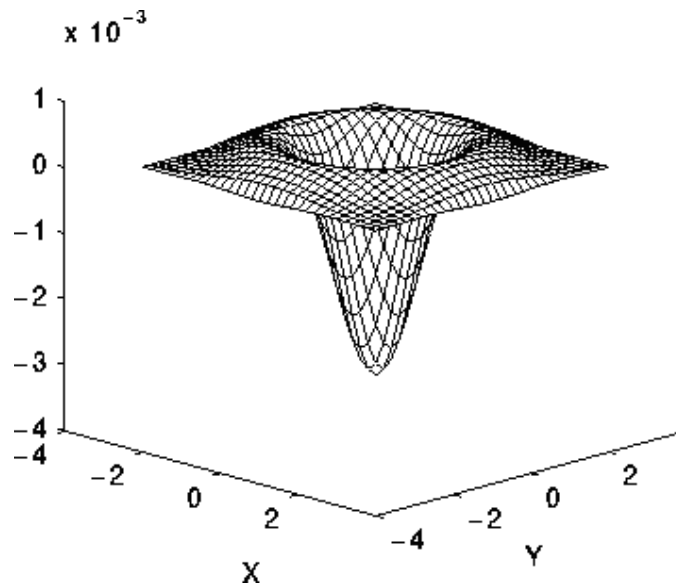
$$\nabla^2 (g(x,y) * f(x,y)) = \nabla^2 (g(x,y)) * f(x,y)$$

Deci, cei doi operatori de convoluție, în loc să fie aplicați succesiv, sunt înlocuiți cu un singur operator, numit “Laplacian of Gaussian”, care poate fi calculat apriori:

$$\text{LoG}(x,y) = \nabla^2 (g(x,y))$$

$$\text{LoG}(x,y) = - (1 / (\pi\sigma^4)) * (1 - (\mathbf{x}^2 + \mathbf{y}^2) / (2 \sigma^2)) * \exp(-(\mathbf{x}^2 + \mathbf{y}^2) / (2 \sigma^2))$$

Funcția este ilustrată în figura următoare:



Funcția Laplacian of Gaussian (LoG) 2D

Aplicarea operatorului consta in convolutia functiei imagine cu o masca de convolutie care aproximeaza in planul discret functia LoG(x,y).

Deoarece intereseaza numai punctele de trecere prin zero, in calculul mastii, constanta $1/(\pi\sigma^4)$ se inlocuieste cu un factor de scalare arbitrar, s. Se observa ca:

$$\text{LoG}(0) = -s$$

.....

$$\text{LoG}(\sigma \sqrt{2}) = 0$$

Deci, functia este nenula pe o distanta circulara de $\sigma \sqrt{2}$ de la origine. Latimea filtrului este de $w = \sigma \sqrt{2} = \text{aproximativ } 3 \sigma$.

Pentru obtinerea efecului dorit, de imbunatatire a fronturilor trebuie sa se utilizeze o masca de marime cuprinsa intre $3w$ si $4w$, ceea ce asigura un echilibru intre marimea mastii si performantele operatorului. Marimea mastii creste odata cu σ , dupa cum rezulta din urmatorul tabel:

| σ | $3w = 6 \sigma 2^{1/2}$ | Marimea mastii |
|----------|-------------------------|----------------|
| 0.5 | 4.24 | 5x5 |
| 1 | 8.48 | 9x9 |
| 2 | 17 | 17x17 |
| 3 | 25.4 | 26x26 |
| 4 | 33.9 | 34x34 |
| 5 | 42.4 | 43x43 |

Urmatoarea masca aproximeaza operatorul LoG pentru $\sigma = 1.4$:

| | | | | | | | | |
|---|---|---|-----|-----|-----|---|---|---|
| 0 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 0 |
| 1 | 2 | 4 | 5 | 5 | 5 | 4 | 2 | 1 |
| 1 | 4 | 5 | 3 | 0 | 3 | 5 | 4 | 1 |
| 2 | 5 | 3 | -12 | -24 | -12 | 3 | 5 | 2 |
| 2 | 5 | 0 | -24 | -40 | -24 | 0 | 5 | 2 |
| 2 | 5 | 3 | -12 | -24 | -12 | 3 | 5 | 2 |
| 1 | 4 | 5 | 3 | 0 | 3 | 5 | 4 | 1 |
| 1 | 2 | 4 | 5 | 5 | 5 | 4 | 2 | 1 |
| 0 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 0 |

Iesirea produsa de operator este o matrice in care apar valori negative si cu parte fractionara. Pentru afisarea sa sub forma unei imagini, valorile din matrice sunt normalizate la domeniul [0-Lmax], cu Lmax =255, de exemplu. In imaginea obtinuta sunt evidentiata frontierele.

LOG poate fi folosit si pentru imbunatatirea frontierelor. Daca iesirea operatorului este adunata la imaginea originala, se obtine o imagine in care zgomotele au fost reduse iar frontierele din imaginea originala sunt evidentiata, au mai mult contrast.

Atunci cand operatorul LoG este utilizat pentru detectia frontierelor, ne intereseaza punctele (x,y) care reprezinta trecerea prin zero a Laplacianului. Detectia acestor puncte este ingreunata de urmatoarele aspecte:

- Prin aplicarea operatorului LoG sunt eliminate punctele izolate de zgomot si structurile mici. Din pacate, punctele de frontiera sunt imprastiate, ceea ce ingreuneaza procesul de detectie a frontierelor.
- Operatorul LoG calculeaza derivata de ordinul 2 a imaginii. Aceasta inseamna ca in zonele in care imaginea are intensitate constanta (gradientul intensitatii

este zero), raspunsul operatorului va fi zero. In vecinatatea unei schimbari de intensitate (frontiere) raspunsul operatorului va fi pozitiv in partea intunecata si negativ in parte luminoasa.

In vecinatatea unei frontiere subtiri dintre 2 regiuni de intensitate uniforma dar diferita, raspunsul operatorului LoG va fi:

- Zero, la o distanta mare de frontiera
- Pozitiv de o parte a frontierei
- Negativ de cealalata parte a frontierei

Pentru detectia punctelor de front din iesirea operatorului LoG se procedeaza ca si in cazul utilizarii Laplacian-ului.

3. Extragerea frontierelor

Urmatorul pas dupa detectia punctelor de front este eliminarea dintre acestea a "falselor" puncte de frontiera si apoi conectarea celor selectate ca puncte de frontiera pentru a forma frontiere (contururi).

3.1. Binarizarea matricei amplitudinilor

Una dintre operatiile uzuale in metodele de extragere a frontierelor este aceea de aplicare a unui prag peste matricea amplitudinilor fronturilor. Operatia de aplicare a unui prag poate fi efectuata si asupra unei matrice imagine, rezultand o imagine binara.

Astfel, dacă $a(k, l)$ este matricea amplitudinilor (sau o matrice imagine), atunci imaginea binară, $B(k, l)$ este :

$$B(k, l) = \begin{cases} 1 & \text{daca } a(k, l) \geq T \\ 0 & \text{altfel} \end{cases}, \text{ unde } T \text{ este o}$$

valoare prag.

Valoarea T poate fi aleasă inspectând histograma imaginii (matricei amplitudine), $a(k, l)$, astfel încât numai un mic procent din elementele matricei a să aibă valoarea $\geq T$. În acest caz, operația de binarizare este globală, deoarece T este ales pe baza unei informații globale și operația se efectuează peste întreaga imagine.

În multe cazuri imaginea (matricea amplitudinilor) conține regiuni cu proprietăți statistice diferite. Binarizarea globală poate produce frontiere subțiri în unele regiuni și late sau întrerupte în altele. În aceste cazuri este de dorit o binarizare adaptată local, adică :

$$B(k, l) = \begin{cases} 1 & \text{daca } a(k, l) \geq T(k, l) \\ 0 & \text{altfel} \end{cases}$$

unde $T(k, l)$ este pragul adaptat local. O metodă de a-l obține constă în calculul mediei aritmetice locale în matricea imagine (sau matricea amplitudinilor produsă de detectorul de fronturi):

$$T(k, l) = \bar{a}(k, l)(1 + p), \text{ unde } p \text{ este un procent care indică nivelul pragului peste media locală.}$$

Deci, pragul diferă de la pixel la pixel în funcție de media amplitudinilor fronturilor din vecinătatea pixelului.

3.2. Detectorul Canny

Detectorul de frontiere Canny este considerat de multi **detectorul optim de frontiere**.

Canny si-a propus definirea unui detector de frontiere care sa imbunatateasca detectoarele existente, din urmatoarele puncte de vedere:

- scaderea ratei de eroare – detectorul sa nu piarda puncte de frontiera si sa nu raspunda la puncte care nu sunt puncte de frontiera
- punctele de frontiera sa fie bine localizate – distanta dintre punctele de frontiera gasite de detector si cele reale sa fie minima
- detectorul sa furnizeze un singur raspuns la un singur punct de frontiera

Pe baza acestor criterii, detectorul Canny efectueaza mai intai o netezire a imaginii, pentru eliminarea zgomotului. Apoi calculeaza Gradientul imaginii pentru a evidentia zonele cu variatie mare a intensitatii. In continuare, algoritmul parcurge aceste zone si elimina orice pixel care nu este un maxim local pe directia gradientului. Matricea gradientului este mai departe redusa prin histerezis. Sunt utilizate doua praguri, $T1 < T2$. Daca valoarea amplitudinii este sub $T1$, pixelul este eliminat; daca valoarea este $>T2$, pixelul este retinut ca pixel de frontiera. Daca valoarea amplitudinii pixelului este cuprinsa intre cele doua praguri, atunci: pixelul este eliminat daca nu exista o cale de la el la un pixel cu amplitudinea gradientului $>T2$, altfel este retinut ca pixel de frontiera.

Algoritmul Canny consta deci din urmatoarii pasi:

Pasul 1:

Se aplica imaginii de intrare un filtru Gaussian.

- Cu cat latimea mastii este mai mare cu atat este mai mica sensibilitatea detectorului la zgomot. Totodata, eroarea de localizare a fronturilor creste odata cu latimea mastii Gaussiene.

Pasul 2:

Se aplica operatorul Sobel imaginii rezultate din pasul 1, obtinandu-se matricea amplitudinilor gradientului.

$$D(x,y) = |Dx| + |Dy|$$

Pasul 3:

Se calculeaza directia gradientului in fiecare punct, obtinandu-se matricea directiilor:

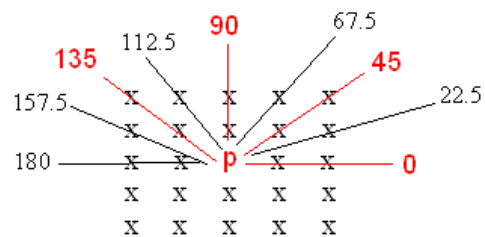
$$\theta(x,y) = \arctg(Dy / Dx), Dx \neq 0$$

$$0, Dy=0$$

$$\pm 90, Dx=0, Dy \neq 0$$

Pasul 4:

Se ajusteaza θ la una dintre directiile din spatiul discret al imaginii, si anume cea mai apropiata de valoarea lui θ . In figura urmatoare sunt desenate cu rosu cele patru directii posibile in spatiul imaginii, din pixelul p:



Directia asignata gradientului in pixelul p se calculeaza astfel:

Directia gradientului = 0 grade , daca $0 \leq \theta < 22.5$ sau $157.5 \leq \theta \leq 180$

= 45 grade, daca $22.5 \leq \theta < 67.5$

= 90 grade, daca $67.5 \leq \theta < 112.5$

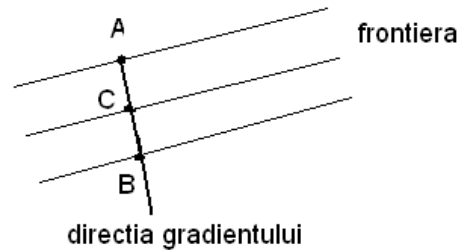
= 135 grade, daca $112.5 \leq \theta < 157.5$

Pasul 5 (non-maxima suppression):

Detectorul Sobel, ca toti detectorii bazati pe masti de convolutie, produce mai multe puncte de front pentru un acelasi punct de frontiera (mai multe masti pot contine un punct de frontiera). De aceea, matricea amplitudinilor poate contine zone late in jurul frontierei.

In acest pas sunt eliminati pixelii care nu au amplitudinea maxima local (in vecinatatea unui pixel).

Sunt declarate puncte de frontiera acele puncte a caror amplitudine este maxima locala pe directia gradientului. Celelalte puncte sunt eliminate (setate la zero).



Pentru fiecare pixel C

Fie A si B pixelii din vecinatatea lui C pe directia gradientului din C

Daca $D(A) > D(C)$ sau $D(B) > D(C)$ atunci $D(C) = 0$

Efectul acestui pas este de subtiere a frontierei fara a o intrerupe. Notam cu I5, matricea amplitudinilor rezultata din acest pas.

Pasul 6 (hysteresis thresholding):

Cu toate ca in primul pas s-a efectuat o filtrare a imaginii, matricea rezultata dupa pasul 5 inca mai contine zgomote, care se manifesta ca pixeli cu contrast mai mic.

In acest pas se elimina pixelii falsi de frontiera (zgomotele) fara intreruperea frontierei (conturului). Pentru aceasta se utilizeaza operatia de "histeresis", care consta in aplicarea a 2 praguri: pragul de jos, P1, si pragul de sus, P2 (P2 aproximativ egal cu $2 \cdot P1$). Aceste praguri se aplica separat imaginii rezultate in pasul anterior, I5, rezultand 2 imagini binare, T1 si T2. In T1 au valoarea 1 pixelii cu amplitudinea $> T1$ iar in T2 aceia cu amplitudinea $> T2$.

Imaginea din T2 are intreruperi in frontiera dar contine mai putine puncte false.

- Punctele din T2 sunt legate in contururi.
- Urmărirea unei frontiere incepe cu un punct din T2. Se conectează puncte din T2 până când se ajunge într-un pixel p care nu mai poate fi conectat cu alt punct din T2 (nu are vecini in T2 pe nici una dintre cele 4 direcții). In acest moment, se caută in T1, in vecinătatea de 8 pixeli a pixelului p, un pixel care poate fi conectat la contur. Se conectează la frontiera puncte din T1 până când se ajunge la un pixel cu valoare diferită de zero in T2 (sau se ajunge la sfârșit de contur). In acest fel, se completează intreruperile de frontiera din T2 cu pixeli din T1.

Metoda histerezis rezolvă problemele care apar la utilizarea unui singur prag.

3.3. Metoda Roberts

Folosește amplitudinea gradientului, calculată prin aplicarea operatorului Roberts.

Fie $D(x,y)$ – matricea amplitudinilor . Algoritmul are 2 etape :

1. Se determină punctele candidat

Pentru aceasta, se examinează **vecinătăți de 4x4 pixeli, disjuncte**. Se determină punctul dintr-o vecinătate cu valoarea maximă a amplitudinii:

$$x_c, y_c \in Vec , D(x_c, y_c) = \max(D(x_i, y_j)), (x_i, y_j) \in Vec$$

Punctul (x_c, y_c) este memorat ca punct candidat dacă:

a) Valoarea sa depășește un anumit prag :

$$D(x_c, y_c) > T$$

b) Nu este un punct de front izolat.

Pentru a determina dacă punctul (x_c, y_c) este izolat se încearcă să se găsească direcția frontierei in (x_c, y_c) prin corelarea valorilor pixelilor din vecinătatea punctului candidat.

- Punctele care aparțin unei linii sau frontiere au aproximativ aceeași amplitudine a frontului. Dacă punctele vecine celui candidat nu satisfac acest criteriu, punctul candidat este considerat punct izolat .

2. Se conectează punctele candidat pe baza direcției frontierei determinate în prima etapă.

- Două puncte candidat sunt conectate dacă se află în vecinătăți adiacente iar diferența dintre valorile lor este sub o valoare de prag.
- Orice punct candidat neconectat este eliminat.

Fiecare pas al algoritmului poate fi paralelizat deoarece se lucrează cu vecinătăți disjuncte.

Algoritmul permite detecția liniilor drepte sau curbe netede. În primul pas, dacă linia prezintă fluctuații puternice, acestea se pot pierde.

3.4. Extragerea frontierelor bazata pe predicția punctului urmator

Fie:

$a(P) = a(x,y)$ amplitudinea frontului in pixelul (x,y)

$\theta(P) = \theta(x,y)$ direcția frontului in pixelul (x,y)

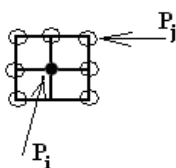
- **Două puncte de front, P_i, P_j , pot fi conectate dacă diferențele în amplitudine și direcție sunt sub valorile de prag date iar amplitudinile fronturilor sunt relativ mari.** Aceste criterii sunt definite prin urmatoarele conditii:

- 1) $| a(P_i) - a(P_j) | \leq T_1$

- 2) $| \theta (P_i) - \theta (P_j) | \leq T_2 \text{ mod } 2\pi$

- 3) $a(P_i) \geq T$ si $a(P_j) \geq T$

Se consideră de regulă vecinătăți de 8 pixeli :



Algoritmul constă din :

- determinarea punctului de start
- determinarea celorlalte puncte prin :
 - predicția următorului punct
 - determinarea următorului punct

Determinarea punctului de start

- Dacă se cunoaște zona de imagine în care se află frontiera, se caută punctul în zona respectivă folosind amplitudinea
- Dacă se cunoaște direcția frontierei, se poate căuta un punct de front cu direcția respectivă
- Altfel : se scanează imaginea până la găsirea a 2 puncte care satisfac criteriile (1) + (2) + (3).

Predicția punctului următor

- Se bazează pe punctele de frontieră deja determinate și eventual pe cunoașterea tipului de frontieră.

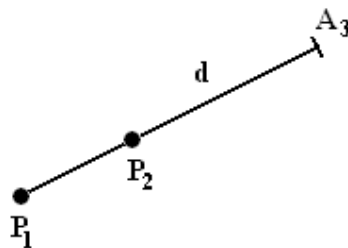
De exemplu, dacă se știe că frontiera este o linie dreaptă, atunci se determină ecuația dreptei pe care se află punctele deja determinate (pe baza metodei celor mai mici pătrate). Următorul punct va fi unul aflat pe aceeași dreaptă, la o distanță d față de ultimul punct determinat. Incrementul d este adesea de un pixel. Dacă se dorește o urmărire mai rapidă se folosește o valoare mai mare a lui d .

Predicția pentru curbe este mai dificilă. Pentru curbe netede, se bazează pe faptul că un segment de curbă scurt este aproximativ liniar.

Determinarea punctului următor

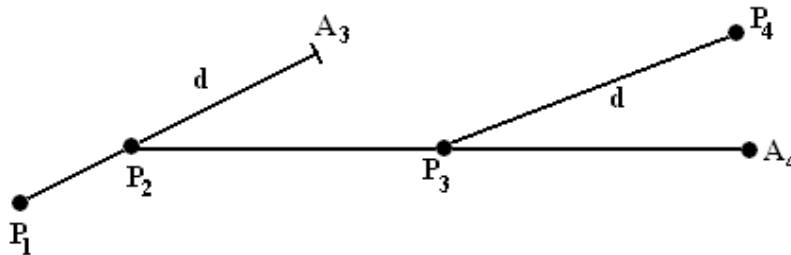
Punctul următor este căutat în vecinătatea punctului determinat prin predicție, pe care-l numim punct anticipat. El trebuie să fie un punct de front care poate fi conectat cu ultimul punct de frontieră determinat, conform condițiilor (1) + (2) + (3).

Fie P_1 și P_2 primele 2 puncte de frontieră, alese conform condițiilor (1) + (2) + (3). Punctul următor anticipat, A_3 , se află la o distanță d de P_2 pe dreapta determinată de $P_1 - P_2$.



A_3 poate să nu fie adresă de pixel sau poate să nu fie un pixel de front.

Punctul de frontieră următor, P_3 , se caută în vecinătatea lui A_3 , astfel încât să fie un punct de front care împreună cu P_2 să satisfacă (1) + (2) + (3):



În continuare, direcția de căutare este dată de $\overline{P_2 - P_3}$.

Dacă distanța d este mare, procesul de urmărire a frontierei este puțin sensibil de zgomot, adecvat pentru curbe netede. Pentru curbe cu zone de curbură mare (fluctuații mari), d trebuie să fie mic.

3.5. Transformata Hough (detectia liniilor drepte)

Inițial, transformata Hough a fost definită pentru detectia liniilor drepte din imagini.

Ulterior a fost extinsă pentru detectia mai multor tipuri de curbe parametrice (cercuri, elipse).

Detectoarele de frontiere prezentate mai înainte extrag frontiere de forma oarecare (prin pixelii de frontiera), fără a furniza informații despre forma geometrică a frontierei. Pasul următor într-un sistem de analiză a imaginilor este extragerea caracteristicilor de nivel coborât din imagine, care presupune aproximarea frontierelor prin primitive geometrice: linii drepte, curbe de anumite tipuri sau curbe generale.

Lesirea transformatei Hough este o matrice binară în care pixelii cu valoarea 1 aparțin unor linii drepte, deci transformata Hough este mai mult decât un detector de frontiere, deoarece furnizează în plus și informația despre forma frontierei.

Intrarea pentru transformata Hough este matricea obținută prin binarizarea matricei amplitudinilor fronturilor.

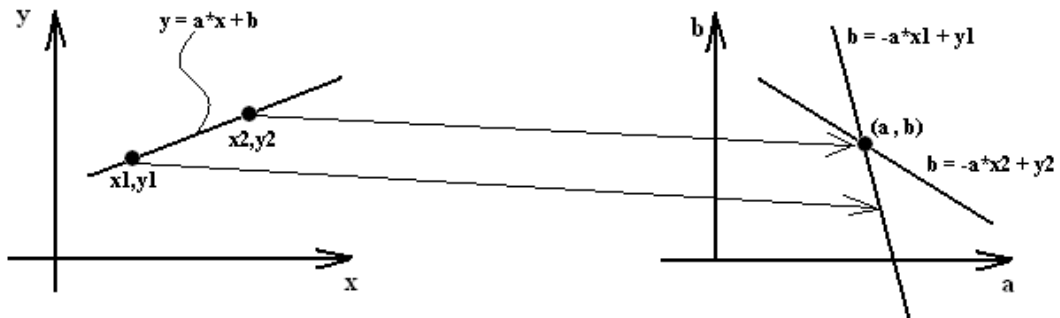
Fie o imagine de $N_1 \times N_2$ pixeli. Cea mai simplă abordare a determinării liniilor drepte este de a găsi linii determinate de perechi de pixeli și apoi a găsi pixelii din imagine care aparțin acestor linii. Numărul maxim de linii posibile este $N(N-1)/2$, unde $N = N_1 \times N_2$. În cazul cel mai defavorabil, fiecare pixel este verificat dacă aparține uneia dintre cele $N(N-1)/2$ linii. Deci, complexitatea calculului este $O(N^3)$, ceea ce face ca metoda să nu poată fi aplicată în practică.

Transformata Hough utilizează o reprezentare parametrică a liniilor pentru reducerea complexității calculului căutării lor în imagini :

$$y = ax + b$$

În spațiul parametric (a, b) fiecare linie este reprezentată printr-un singur punct (a, b) .

Fie o linie care trece prin 2 puncte $(x_1, y_1), (x_2, y_2)$:



Liniile care trec prin (x_1, y_1) se reprezintă în spațiul parametric prin linia $b = -a*x_1 + y_1$ iar cele care trec prin (x_2, y_2) se reprezintă prin linia $b = -a*x_2 + y_2$. Intersecția acestor 2 linii din spațiul parametric determină în mod unic dreapta care trece prin $(x_i, y_i), i = 1, 2$.

Pentru detecția liniilor drepte se poate proceda astfel :

1. Se discretizează spațiul parametric și se reprezintă printr-o matrice P, astfel încât $P[i][j]$ corespunde unui punct (a_j, b_i)

$$a_1 \leq a \leq a_K, a_j = a_1 + j * da$$

$$b_1 \leq b \leq b_L, b_i = b_1 + i * db$$

Se inițializează elementele matricei P cu zero.

2. Pentru fiecare pixel (x, y) , cu $B(x, y) = 1$

// se formează ecuația $b = -a*x + y$

pentru fiecare $a_1 \leq a_m \leq a_K // (a_m = a_1 + m * da)$

$$b_m = -a_m * x + y$$

$$i = (b_m - b_1) / (b_L - b_1) * (L - 1)$$

$$P[i][m] ++;$$

3. Pentru fiecare element al matricei P, $P[i][j]$

dacă $P[i][j] \geq T$ atunci

// in imagine exista dreapta $y = a*x + b$, unde $a = a_1 + j*$ da, $b = b_1 + i*db$

$P[i][j]$ – numără intersecțiile liniilor care trec prin (a , b) și indică numărul de pixeli care satisfac ecuația

$$y = a*x + b.$$

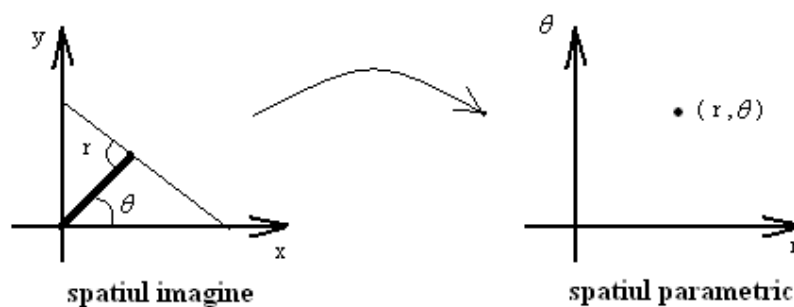
Precizia metodei depinde de pasul de discretizare a spațiului (a , b), deci de mărimea matricei P.

Complexitatea calculului transformatei Hough este $O(k*N)$, unde k este numărul de subintervale pe axa a iar N este numărul de pixeli din imaginea de intrare.

Utilizarea ecuației parametrice $y = a*x + b$ produce dificultăți în reprezentarea liniilor verticale ($a \rightarrow \infty$).

O reprezentare mai bună este reprezentarea polară :

$$r = x * \cos(\theta) + y * \sin(\theta)$$



- O linie care trece prin (x_1, y_1) reprezintă o curbă sinusoidală, $r = x_1 \cdot \cos(\theta) + y_1 \cdot \sin(\theta)$ în spațiul parametric (r, θ) .
- Puncte colineare, (x_i, y_i) , din imagine, corespund intersecțiilor sinusoidelor din spațiul parametric.

Deci, **punctele de intersecție definesc liniile drepte din imagine.**

Pentru o imagine de $N_1 \times N_2$ pixeli,

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}, \quad -\sqrt{N_1^2 + N_2^2} \leq r \leq \sqrt{N_1^2 + N_2^2} .$$

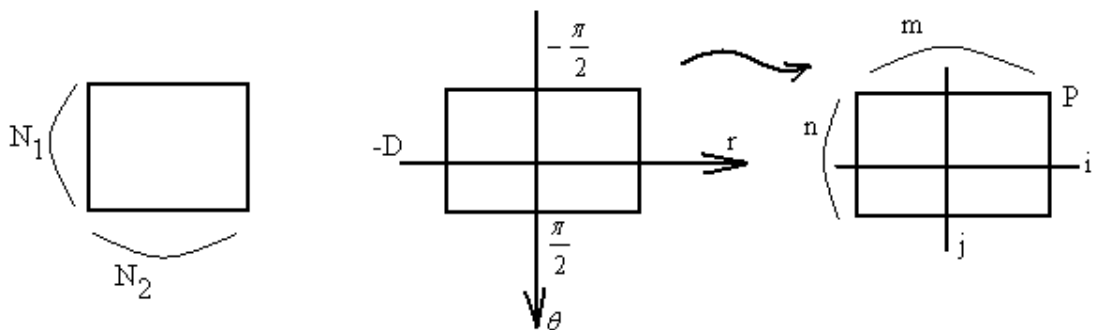
Discretizarea spațiului parametric :

- intervalul $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ se împarte în n subintervale;

- intervalul $[-D, D]$ se împarte în m subintervale, unde

$$D = \sqrt{N_1^2 + N_2^2}$$

Fie $P[n][m]$ matricea în care se memorează punctele grilei spațiului parametric.



pentru $\begin{matrix} r = -D, j = 0 \\ r = D, j = m - 1 \end{matrix} \Rightarrow j = (\text{int})\left(0.5 + \frac{r + D}{2D}(m + 1)\right)$

$$\text{pentru } \begin{cases} \theta = -\frac{\pi}{2}, i = 0 \\ \theta = \frac{\pi}{2}; i = n - 1 \end{cases} \Rightarrow \theta = \left(\frac{i * \pi}{n - 1} - \frac{\pi}{2} \right)$$

Se precalculeaza valorile funcțiilor *sin* și *cos* pentru valorile discrete ale lui θ :

```
float *COS, *SIN, t;
for ( i=0; i<n; i++)
{
    t = ( i*M_PI/(n-1) - M_PI_2 );
    COS[i] = cos(t);
    SIN[i] = sin(t);
}
```

Implementarea transformatei Hough

```
void Hough(Imagine B, //matricea binară a amplitudinilor
           int P[N][M],
           int n, int m,
           int N1, int N2,
           float *COS, float *SIN,
           int T,
           Imagine out)
{.....
    float D = sqrt((float) N1*N1 + (float) N2*N2);
    for (k = 0; k < n; k++)
        for(l = 0; l < m; l++)
            P[k][l] = 0;
    for (k = 0; k < N1; k++)
        for(l = 0; l < N2; l++)
        {
            if (B[k][l] == 1)
```

```

    {
        for(i = 0; i < n; i++)//pentru fiecare  $\theta$ 
        {
            r = l * COS[i] + k * SIN[i];
            j = (r + D) * (m - 1) / (2 * D) + 0.5;
            P[i][j] ++;
        }
    }
}
//Se calculează transformata Hough inversă (din spațiul (r ,  $\theta$ ) în (x , y))
//Pentru fiecare element P[k][l] > T, se înscrie în matricea out, sub formă de
//pixeli = 1, linia dreaptă corespunzătoare elementului (k , l)
//Se efectuează operația and între rezultatul transformatei și matricea de intrare

```

```

for(k = 0 ; k < n ; k++)
for(l = 0 ; l < m ; l++)
{
    if ( P[k][l] > T )
    {
        r = (2.0 * l * D) / (m - 1) - D;
        // se marcheaza in matricea out pixelii care satisfac ecuatia
        // r = x * COS[k] + y * SIN[k]
        for(i = 0 ; i < N2 ; i++)
        {
            if ( SIN[k] == 0 )
                y = N1 / 2;
            else
                y = ( r - i * COS[k] ) / SIN[k];
            j = y + 0.5;
            if ( j >= 0 && j < N1 )
                if ( B[j][i] == 1 )
                    out[j][i] = 1;
        }
    }
}

```

}
 }
 }

Anumiți detectori de fronturi produc nu numai mărimea frontului ci și direcția frontului, $\Phi(x, y)$.

- Informația direcțională poate fi utilizată pentru a reduce calculele în transformata Hough.

Astfel, diferențiind în raport cu x ambele părți ale ecuației :

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

rezulta:

$$\frac{dr}{dx} = \cos(\theta) + \frac{dy}{dx} \cdot \sin(\theta) = 0 \Rightarrow \frac{dy}{dx} = -\frac{\cos(\theta)}{\sin(\theta)} = -\cot(\theta) = \operatorname{tg}\left(\theta + \frac{\pi}{2}\right)$$

Direcția frontului in (x , y) este $\phi_{x,y} = \operatorname{arctg}\left(\frac{dy}{dx}\right)$

$$\operatorname{tg}(\phi) = \frac{dy}{dx}$$

Deci :

$$\phi = \frac{\pi}{2} + \theta \Rightarrow \theta = \phi - \frac{\pi}{2}$$

Rezultă că pentru fiecare pixel $B[k][l] = 1$, numai un singur element al matricei P trebuie să fie actualizat : $P[i_\theta][j_\theta]$, unde :

$\theta = \operatorname{FI}[k][l]$, - FI este matricea direcției frontului

$$i_\theta = (\theta/\pi + 0.5)(n-1)$$

$$r_\theta = l \cdot \operatorname{COS}[i_\theta] + k \cdot \operatorname{SIN}[i_\theta]$$

$$j_\theta = (\operatorname{int}) r_\theta$$

Complexitatea calculului transformatei Hough se reduce in acest caz de la $O(k \cdot N)$ la $O(N)$.

Transformata Hough se poate generaliza pentru a detecta orice curbă parametrică. Metoda este practică numai pentru curbe cu un număr mic de parametri. De exemplu, pentru un cerc cu ecuația $(x-a)^2 + (y-b)^2 = r^2$, parametrii sunt r, a, b . Este necesară o matrice P tridimensională, $P(r, a, b)$.