

3 Detecția dreptelor cu transformata Hough

3.1 Obiective

Obiectivul principal al acestei lucrări este studierea transformatei Hough pentru detecția dreptelor în imagini binare.

3.2 Fundamente teoretice

Transformata Hough este o metodă care rezolvă o problemă clasică din viziunea artificială: găsirea dreptelor într-o imagine binară ce conține o mulțime de puncte de interes (de exemplu puncte de muchii). Metoda directă de a calcula drepte din fiecare pereche de puncte are o complexitate computațională ridicată de $O(n^2)$, și nu este aplicabilă pentru un număr mare de puncte. Transformata Hough a fost propusă și patentată de Peter Hough [1], și în varianta inițială, a fost o metodă de timp real pentru a număra câte puncte sunt plasate pe fiecare dreaptă posibilă dintr-o imagine. Această metodă se bazează pe reprezentarea dreptei sub formă pantă-termen liber ($y=ax+b$), și pe construirea unui spațiu parametric, numit și acumulator Hough. Pentru fiecare punct de interes din imagine se calculează toate dreptele posibile care trec prin el, și se incrementează elementele din spațiul parametric. Dreptele relevante sunt localizate în maximele locale ale spațiului parametric.

Această reprezentare este sub-optimală, deoarece nu este mărginită. Pentru a reprezenta toate dreptele posibile din imagine, panta și termenul liber trebuie să varieze în domeniul $-\infty$ și $+\infty$. Modificările propuse de Duda și Hart [2] au făcut transformata Hough populară în domeniul viziunii artificiale. Principala problemă legată de parametri nemărginiți a fost rezolvată prin parametrizarea normală a dreptei în sistemul de coordonate polar. Parametrizarea normală a unei drepte constă în reprezentarea dreptei prin vectorul normal (perpendicular) și distanța de la origine. Reprezentarea normală se mai numește și reprezentarea ρ - θ (Figura 3.1).

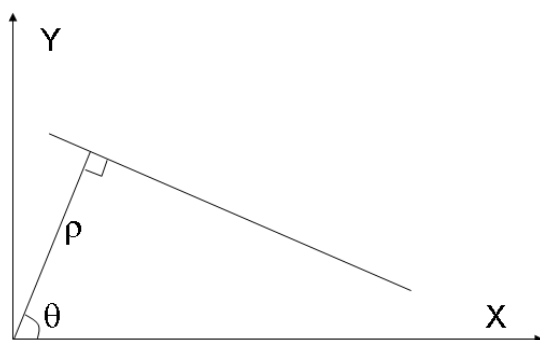


Figura 3.1. Reprezentarea dreptei în sistemul de coordonate polar. Dreapta se află la o distanță ρ față de origine și normala face unghiul θ cu axa Ox.

O dreaptă poate fi reprezentată folosind parametrii ρ și θ :

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (1)$$

Parametrii ρ și θ pot avea o infinitate de valori. Pentru a descrește complexitatea

computațională se introduce cuantizarea parametrilor ρ și θ , proces prin care se constrâng parametrii să aibă un număr finit de valori. Parametrii ρ și θ au un interval de variație limitat deoarece imaginea are o dimensiune finită. Valoarea maximă pentru ρ este diagonală imaginii. În funcție de intervalul ales pentru θ , există două configurații echivalente pentru domeniul parametrilor. Prima este cea propusă în articolul original iar noi o vom adopta pe a doua.

$$\begin{aligned} 1. \theta &\in [-90^\circ, 90^\circ) \text{ or } \theta \in [0, 180^\circ), \rho \in [-\rho_{\max}, +\rho_{\max}] \\ 2. \theta &\in [0, 360^\circ), \rho \in [0, +\rho_{\max}] \end{aligned} \quad (2)$$

De asemenea, pentru fiecare dintre cei doi parametri ai dreptei se stabilește un nivel de cuantizare care depinde de acuratețea cerută (de exemplu: pasul pentru ρ poate fi de 10, 1, 0.5 pixeli etc., iar pasul pentru θ poate fi de 10 grade, 1 grad, 0.5 grade etc.).

În general printr-un punct din planul imagine (x_0, y_0), putem defini un set de drepte care trec prin punctul respectiv la diferite unghiuri θ . Având (x_0, y_0) și θ , parametrul dreptei ρ se poate calcula conform ecuației (2). Pentru a detecta drepte dintr-o imagine, algoritmul Hough folosește o schemă de votare: fiecare punct de interes (x_0, y_0), de exemplu pixelii de pe muchii, votează pentru toate dreptele cu parametrii (ρ, θ) care trec prin punctul respectiv. Dreptele care primesc cele mai multe voturi sunt considerate dreptele finale detectate de algoritmul Hough. Un exemplu este prezentat în Figura 3.2.

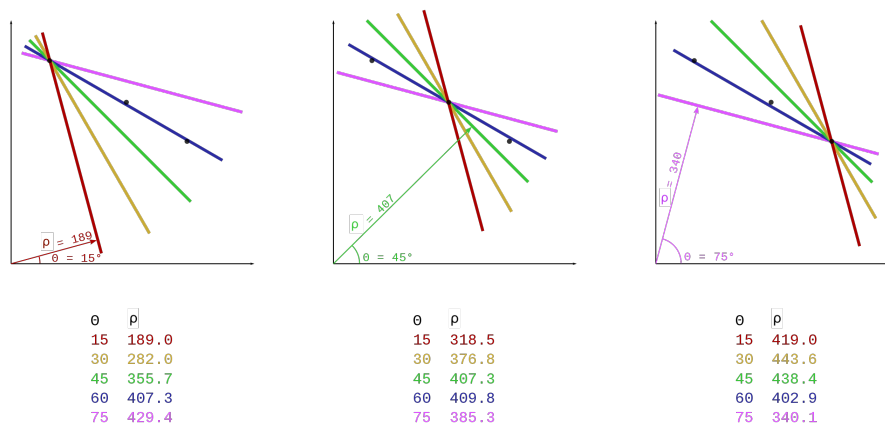


Figura 3.2 Se consideră 3 puncte de interes marcate cu negru. Se dorește determinarea dreptelor din imagine. Pentru fiecare punct avem un set de drepte având parametrii (ρ, θ) care trec prin acel punct. Fiecare punct votează pentru dreptele care trec prin acel punct. Observăm că în toate cazurile, pentru dreapta cu $\theta=60^\circ$, valorile pentru ρ sunt similare. Se poate concluziona că punctele se află în vecinătatea dreptei albastre cu $\theta=60^\circ$, care va fi detectată de algoritmul Hough. Exemplu preluat din [4].

Pentru a stoca voturile, algoritmul Hough folosește un vector 2D (matrice), denumit acumulatorul Hough (H). Acesta reprezintă spațiul parametrilor (ρ, θ) cuantizați ai dreptei. Pașii de cuantizare pentru ρ și θ sunt $\Delta\rho$ și $\Delta\theta$, respectiv. Valorile lor maxime sunt ρ_{\max} și θ_{\max} . Atunci acumulatorul va avea o dimensiune de $(\rho_{\max}/\Delta\rho \times \theta_{\max}/\Delta\theta)$. Operația de incrementare a unei locații Hough poate fi ponderată – de exemplu cu modulul gradientului. După ce am construit acumulatorul, dreptele relevante se extrag ca maxime locale ale acestuia. Un maxim local este un punct unde valoarea din acumulator este mai mare decât toate valorile dintr-o vecinătate (pătratică). Algoritmul Hough este următorul:

Algoritmul Hough

1. Se inițializează fiecare celulă din H cu 0
 2. Se creează acumulatorul H :
Pentru fiecare punct de muchie $P(x, y)$ din imagine
Pentru fiecare θ de 0 la θ_{\max} (cu un pas de $\Delta\theta$)
Se calculează $\rho = x \cos(\theta) + y \sin(\theta)$
Dacă $\rho \in [0, +\rho_{\max}]$ se incrementează $H(\rho, \theta)$
 3. Se găsesc primele k maxime locale din H :
Se verifică dacă $H(\rho, \theta)$ este maxim local într-o fereastră $n \times n$ centrată pe celula respectivă:
Pentru fiecare θ de la n la $\theta_{\max} - n$ (cu un pas de $\Delta\theta$)
Pentru fiecare ρ de la n la $\rho_{\max} - n$ (cu un pas de $\Delta\rho$)
Se stochează tripletul $[\rho, \theta, H(\rho, \theta)]$ dacă $H(\rho, \theta)$ are cea mai mare valoare din fereastra $n \times n$ și $H(\rho, \theta) > T$
 4. Se ordonează descrescător tripletele stocate în funcție de $H(\rho, \theta)$ și se returnează primele k drepte (ρ, θ)
-

Parametrii algoritmului Hough sunt următorii:

- P - o imagine binară
- θ_{\max} - valoarea maximă pe care o poate lua θ
- ρ_{\max} - valoarea maximă pe care o poate lua ρ
- $\Delta\theta$ - pasul pentru θ
- $\Delta\rho$ - pasul pentru ρ
- n - dimensiunea ferestrei pentru determinarea maximului local
- T - numărul minim de voturi (sau echivalent numărul minim de puncte) pentru ca o dreaptă să fie considerată validă
- k - numărul de drepte dorit

Un exemplu de detecție a liniilor pe baza transformatei Hough se prezintă în Figura 3.3. Domeniul de variație al parametrilor pentru acest exemplu este $[0, 360)$ grade pentru θ , și $[0, 144]$ pixeli pentru ρ . Pasul parametrilor este de 1 grad pentru θ și de 1 pixel pentru ρ .

Alegerea unui nivel de cuantizare adecvat este foarte importantă. Dacă se face o cuantizare prea fină, rezoluția crește odată cu timpul de procesare, și cresc și șansele ca puncte aparent colineare să incrementeze celule diferite din acumulator (acest lucru va cauza detecții multiple ale aceleiași drepte, sau fragmentarea unei drepte în mai multe părți).

Deși transformata Hough se folosește cel mai des pentru detecția dreptelor, ea poate fi folosită și pentru detecția curbilor mai complexe, atât timp cât o parametrizare adecvată este disponibilă. Duda și Hart [2] au propus detecția cercurilor, folosind un spațiu de parametri tridimensional și transformând fiecare punct într-un con circular în spațiul parametric (toate cercurile posibile ce conțin respectivul punct). Mai târziu, Ballard a generalizat transformata Hough pentru a detecta orice formă non-analitică [3].

3.3 Detalii de implementare

Folosiți cea mai simplă configurație posibilă pentru cuantizarea parametrilor: 1 pixel pentru ρ și 1 grad pentru θ . Folosiți a doua variantă pentru domeniul de variație al parametrilor conform ecuațiilor (3). Astfel vom detecta dreptele care au unghiul

$\theta \in [0, 360^\circ)$ cu un pas $\Delta\theta=1$ și $\rho \in [0, \rho_{max}]$ cu un pas $\Delta\rho=1$. Dimensiunea acumulatorului Hough va fi de $\rho_{max}+1$ rânduri și $\theta_{max}=360$ de coloane, unde ρ_{max} este diagonala imaginii:

$$\rho_{max} = \sqrt{height^2 + width^2}$$

Se declară acumulatorul Hough ca o matrice cu elemente de tip întreg:

```
Mat Hough ( $\rho_{max}+1$ , 360, CV_32SC1);
```

Acumulatorul se inițializează cu 0 folosind:

```
Hough.setTo(0);
```

Pentru a calcula ρ conform ecuației (2) este necesară transformarea θ din grade în radiani:

$$\theta_{rad} = \theta * CV_PI/180$$

Apoi se poate calcula ρ :

$$\rho = x \cos(\theta_{rad}) + y \sin(\theta_{rad})$$

Acumulatorul se modifică (numai dacă $\rho \in [0, +\rho_{max}]$) folosind:

```
Hough.at<int>( $\rho$ ,  $\theta$ )++;
```

Pentru a afișa acumulatorul sub forma unei imagini cu nivele de gri, acesta trebuie normalizat ca valorile să fie între 0-255. Se poate folosi funcția `normalize` din biblioteca OpenCV sau se poate normaliza pe baza valorilor maxime din acumulator folosind secțiunea de cod următoare:

```
Mat houghImg;  
Hough.convertTo(houghImg, CV_8UC1, 255.f/maxHough);
```

unde `maxHough` reprezintă valoarea maximă din acumulator după construire. Calculele ulterioare se fac pe acumulatorul original.

Pentru a localiza maximele locale din acumulator, se va testa pentru fiecare element dacă este maxim local într-o fereastră patrată ($n \times n$) centrată pe element. Un element este maxim local dacă este mai mare decât toate elementele din fereastră. Rețineți acele elemente care sunt maxime locale și care au valoarea mai mare decât un prag. Ordoneți elementele reținute, și păstrați primele k vârfuri, care reprezintă cele mai mari maxime locale și corespund la dreptele importante.

Următoarea structură vă permite stocarea maximelor locale și sortarea lor cu un apel la metoda `sort` din biblioteca `algorithm`. Operatorul `<` a fost redefinit pentru a realiza sortarea descrescătoare a maximelor locale în funcție de valorile din acumulator (hval).

```
struct peak{
    int theta, ro, hval;
    bool operator < (const peak& o) const {
        return hval > o.hval;
    }
};
```

3.4 Activitate practică

1. Calculați acumulatorul Hough folosind imaginea de muchii. Afișați rezultatul sub forma unei imagini cu niveluri de gri.
2. Găsiți primele k maxime locale. Folosiți mărimi diferite pentru fereastra de suport, de exemplu de 3×3 , 7×7 sau 11×11 (parametru pentru metodă) și diferite valori pentru pragul T .
3. Desenați liniile asociate cu aceste k vârfuri atât pe imaginea originală cât și pe imaginea de muchii.

3.5 Exemple de rezultate

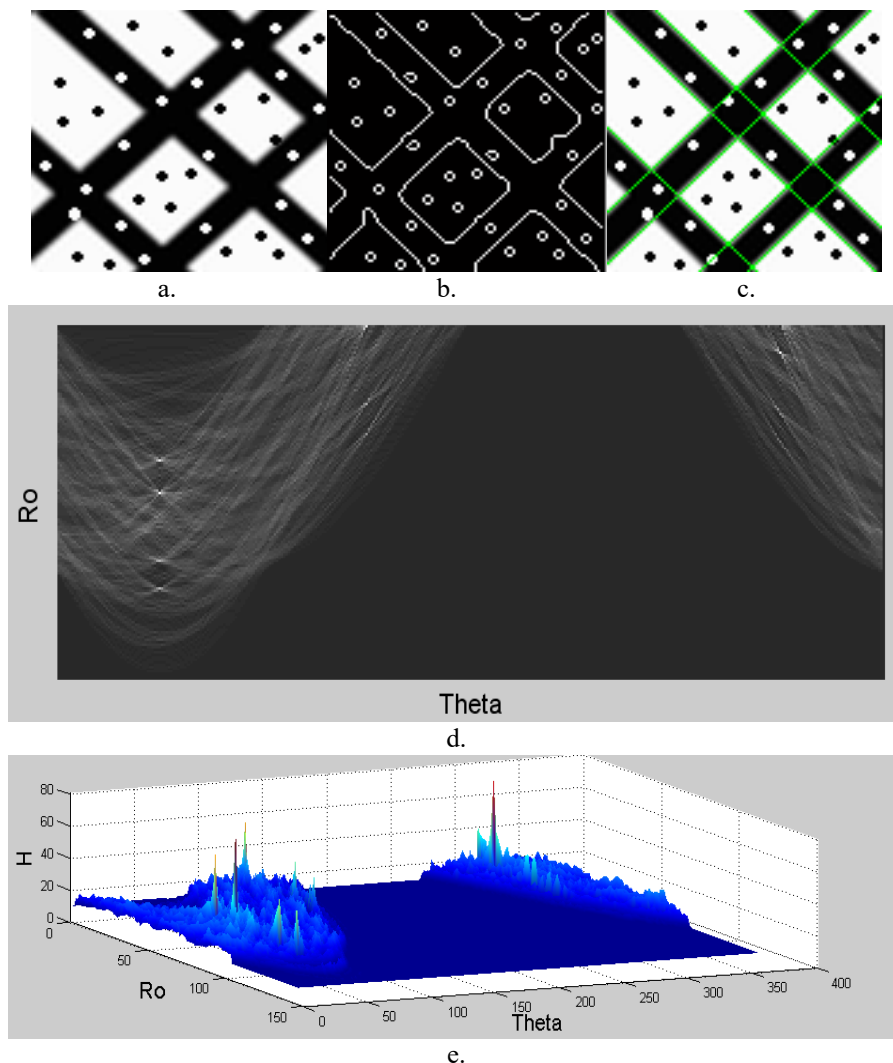


Figura 3.3. **a.** Imagine cu un model cu muchii drepte, afectată de zgomot sare și piper **b.** Muchiile detectate cu un detector de muchii Canny, **c.** Cele mai relevante drepte sunt marcate cu verde, și ele se vor asocia cu cele mai

relevante 8 maxime din acumulatorul Hough, **d.** Acumulatorul Hough afișat sub forma unei imagini cu niveluri de gri, **e.** Acumulatorul Hough afișat în 3D, folosind codificarea în culoare.

3.6 Referințe

- [1] P. Hough, "Method and means for recognizing complex patterns", US patent 3,069,654, 1962.
- [2] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM*, Vol. 15, pp. 11–15, 1972.
- [3] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", *Pattern Recognition*, Vol.13, No.2, p.111-122, 1981.
- [4] https://en.wikipedia.org/wiki/Hough_transform