

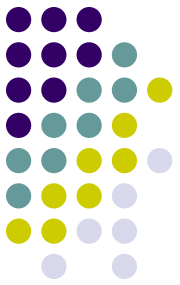
Paralelizarea transformatei HOUGH

Ionescu Gabriel
GMRV



Cuprins

- Introducere
- Tehnica de paralelizare
- Realizarea hardware
- Concluzii





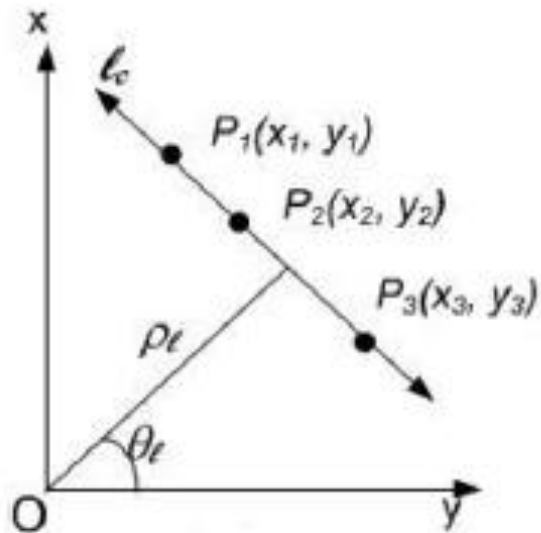
Introducere

- Transformata Hough (TH) este folosita in procesarea imaginilor si detectarea formelor
- Se bazeaza pe reprezentarea parametrica a liniilor ce trec printr-un punct al planului:
 - Un punct $P(x,y)$ este definit de un set de linii care se intersecteaza. Fiecare linie se scrie sub forma:

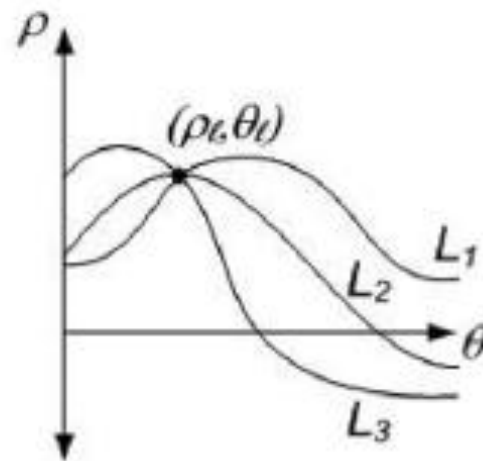
$$\rho = x * \cos(\theta) + y * \sin(\theta)$$

- Unde cei doi paramteri (unici) sunt reprezentati intr-un plan ce reprezinta rezultatul aplicarii transformatei.

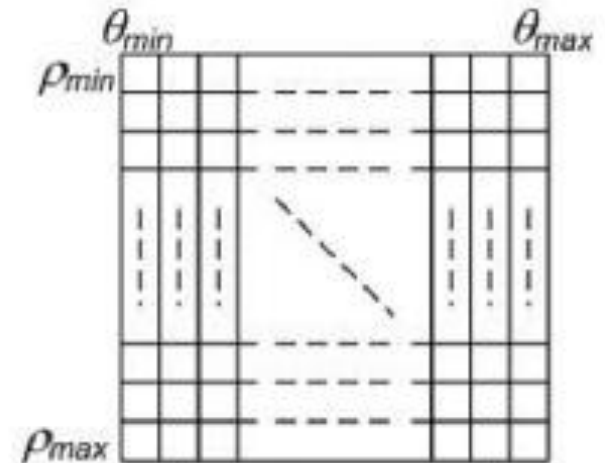
Introducere



(a)



(b)



(c)

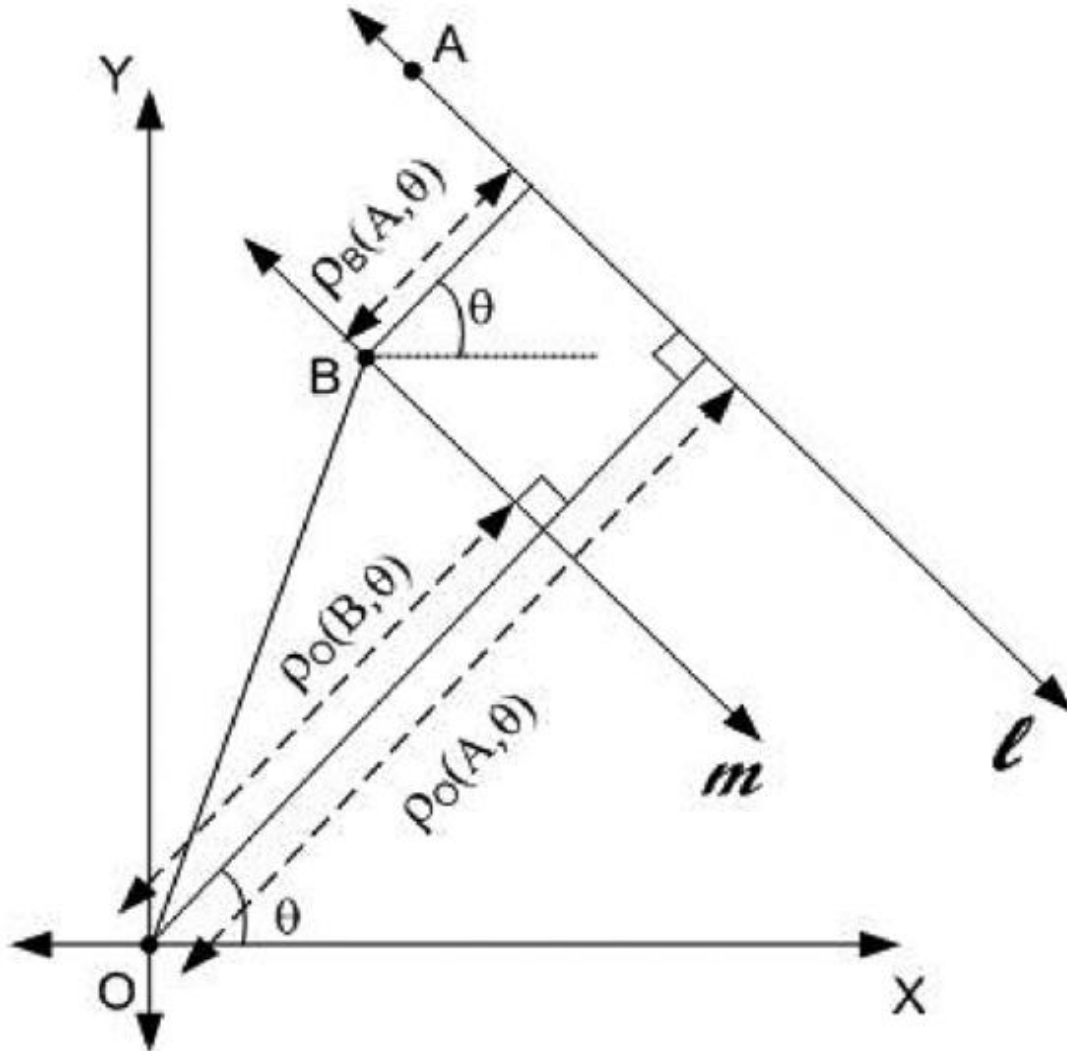
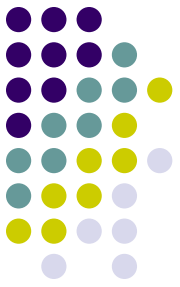
- (a) – figura cu 3 puncte colineare
- (b) - figura ce rezulta dupa aplicarea TH
- (c) - reprezentare cu un array 2D

Tehnica de paralelizare



- Paralelizarea transformatei Hough se bazeaza pe o proprietate importanta - proprietatea de aditivitate:
 - Transformata Hough a unui punct A , relativ la originea O , este egala cu suma transformatelor de la punctul A la punctul B si de la punctul B la punctul O

Tehnica de paralelizare



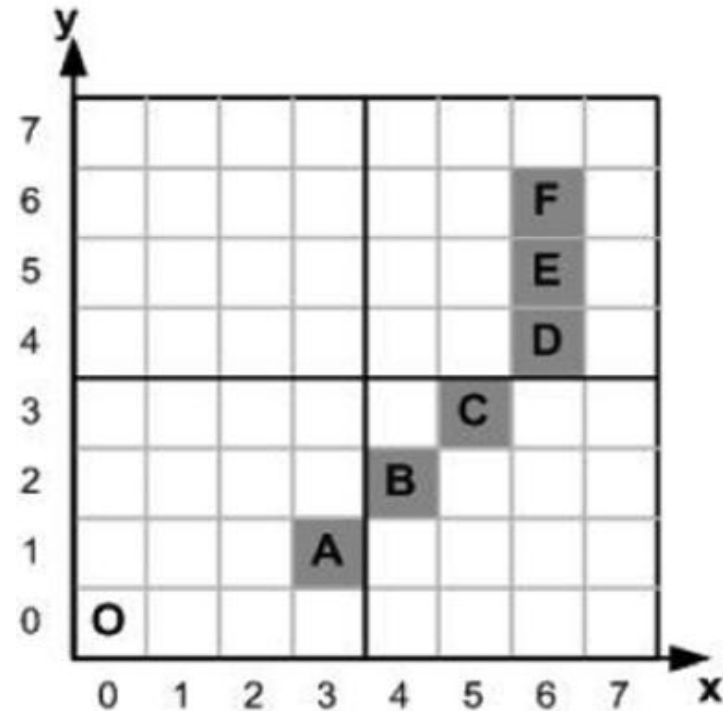
Din figura se
poate deduce :
 $HT(A,O) = HT(A,B) +$
 $+ HT(B,O)$



Aplicarea pe o imagine

- Pentru a putea aplica proprietatea enuntata mai devreme se considera o imagine de $M \times M$ impartita in $K \times K$ zone. De exemplu:

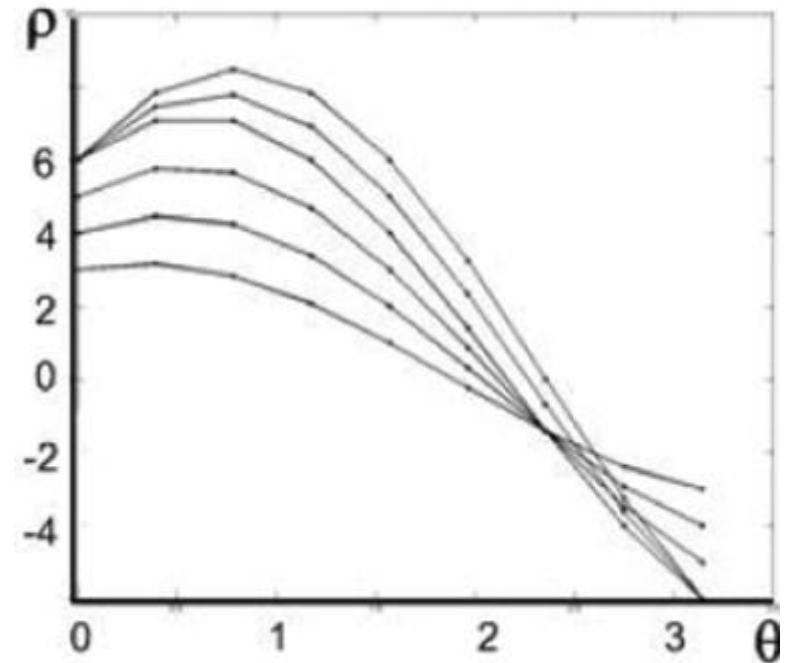
Transformata pentru imaginea alaturata:





Aplicarea pe o imagine

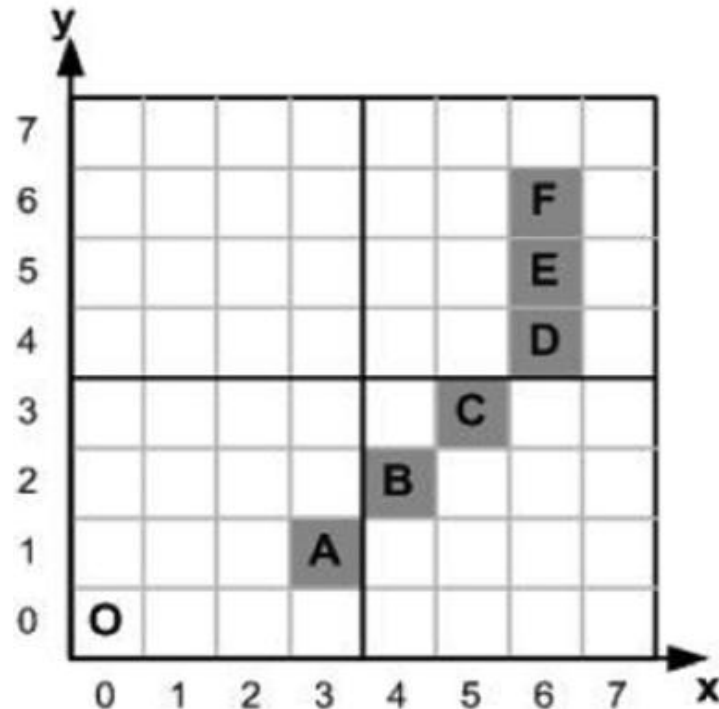
- Rezultatul pentru imaginea din slide-ul anterior:
- 4 linii se intalnesc la $3\pi/4$ si in π sau 0
- Intersectiile reprezinta liniile ABCD, respectiv DEF





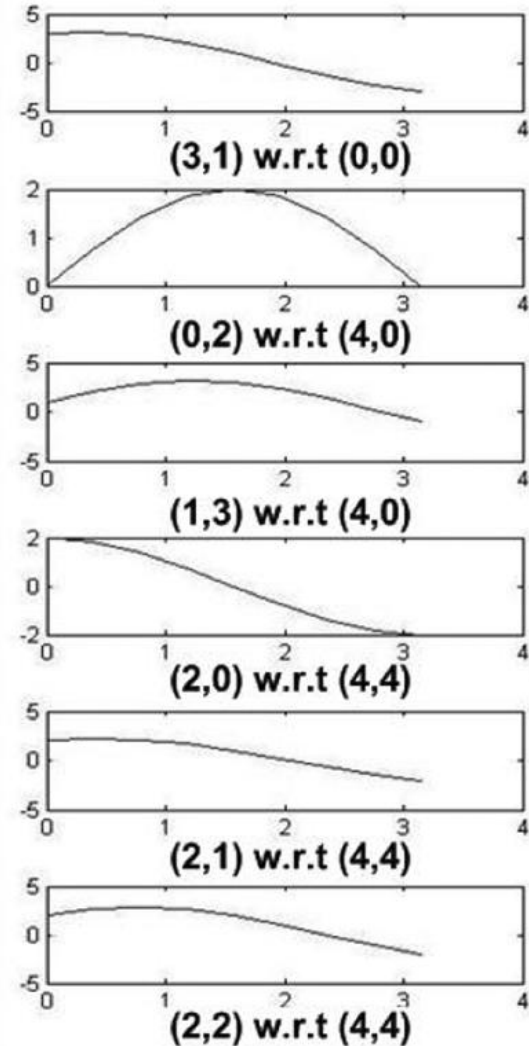
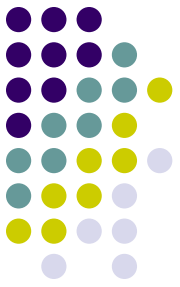
Aplicarea pe o imagine

- Consideram aceeași imagine de 8×8 împartită în 4 blocuri de 4×4 și aplicăm TH respectând proprietatea de aditivitate.



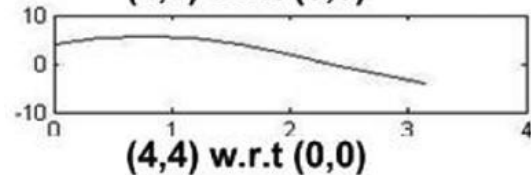
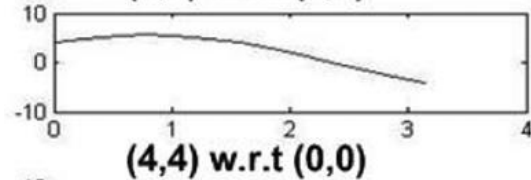
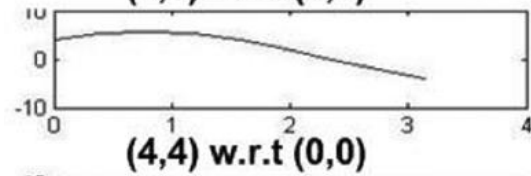
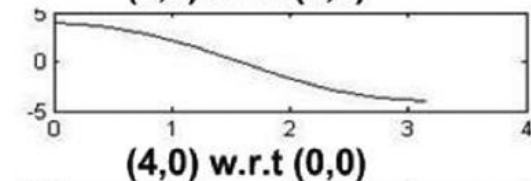
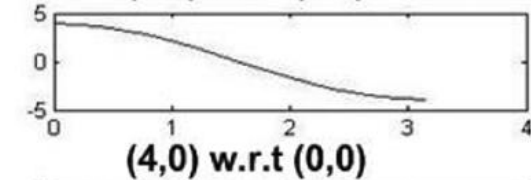
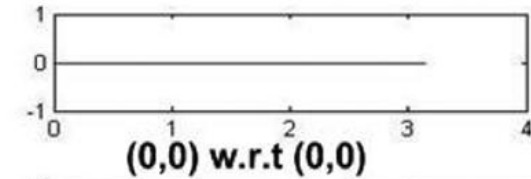
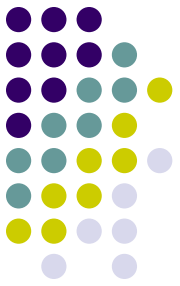
Aplicarea pe o imagine

- Transformarea Hough locala pentru zonele A,B,C,D,E,F:



Aplicarea pe o imagine

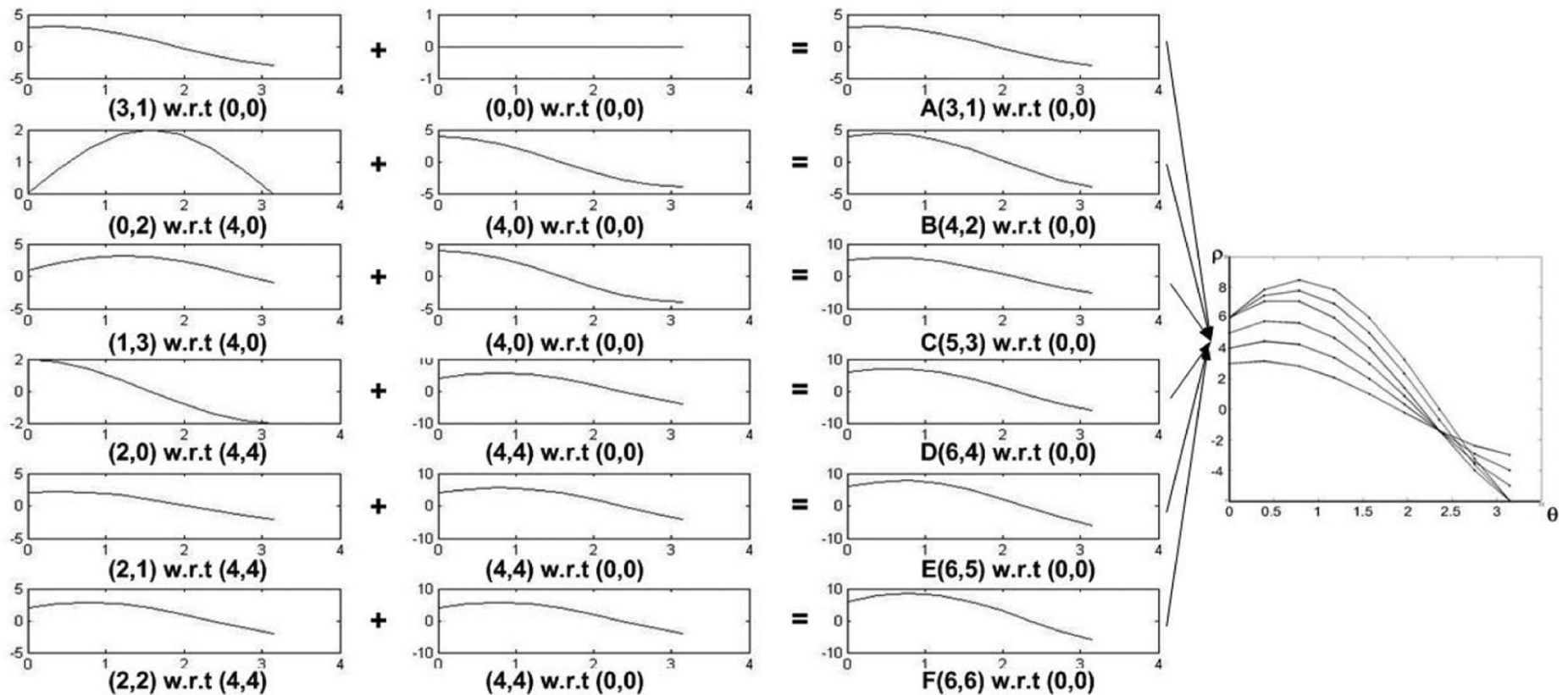
- Transformarea Hough locala pentru originea fiecarui dreptunghi:



Aplicarea pe o imagine



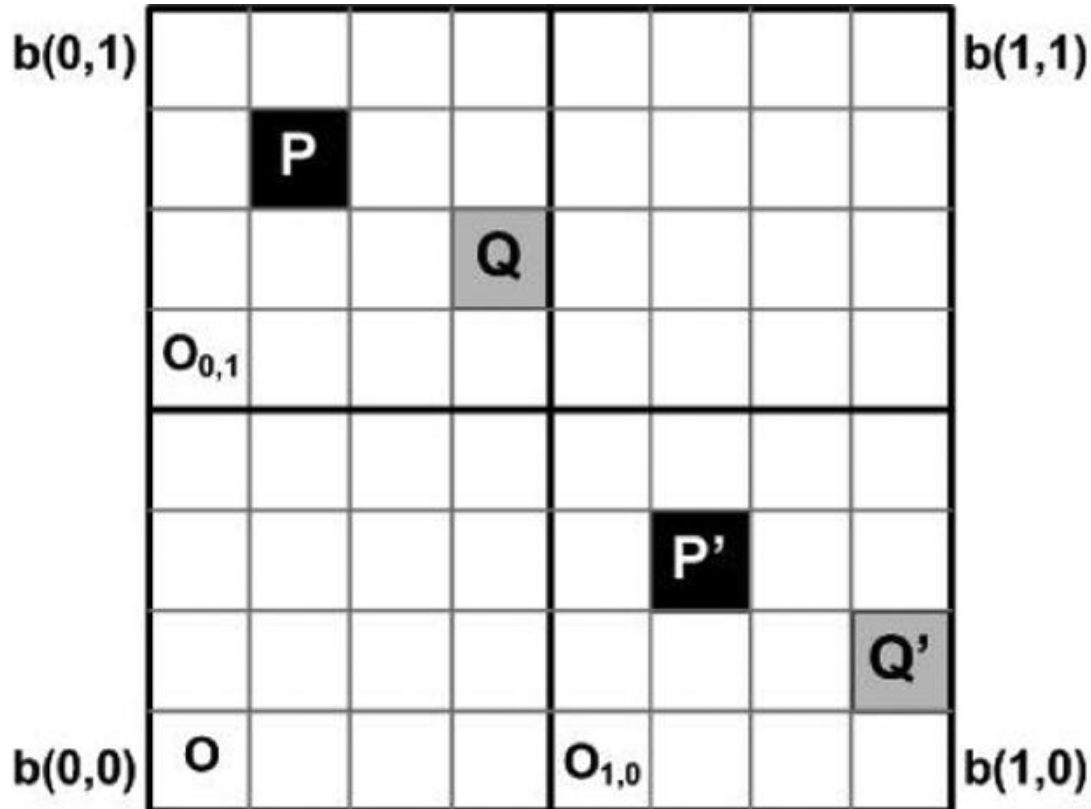
- Rezultatul final:





Tehnica de paralelizare

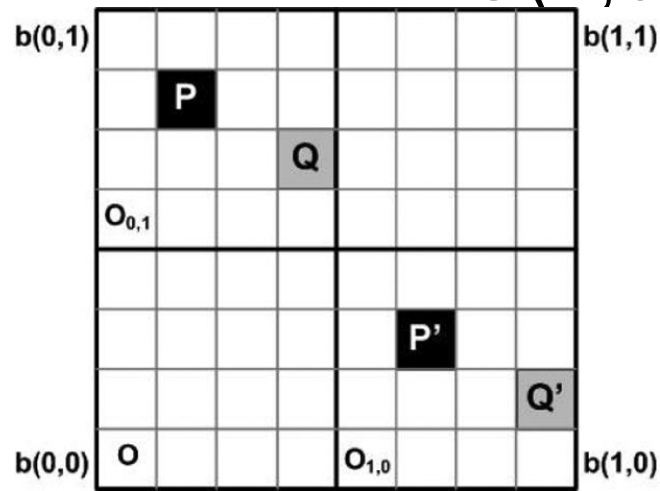
- Consideram o imagine de 8x8 impartita in 4 blocuri.





Tehnica de paralelizare

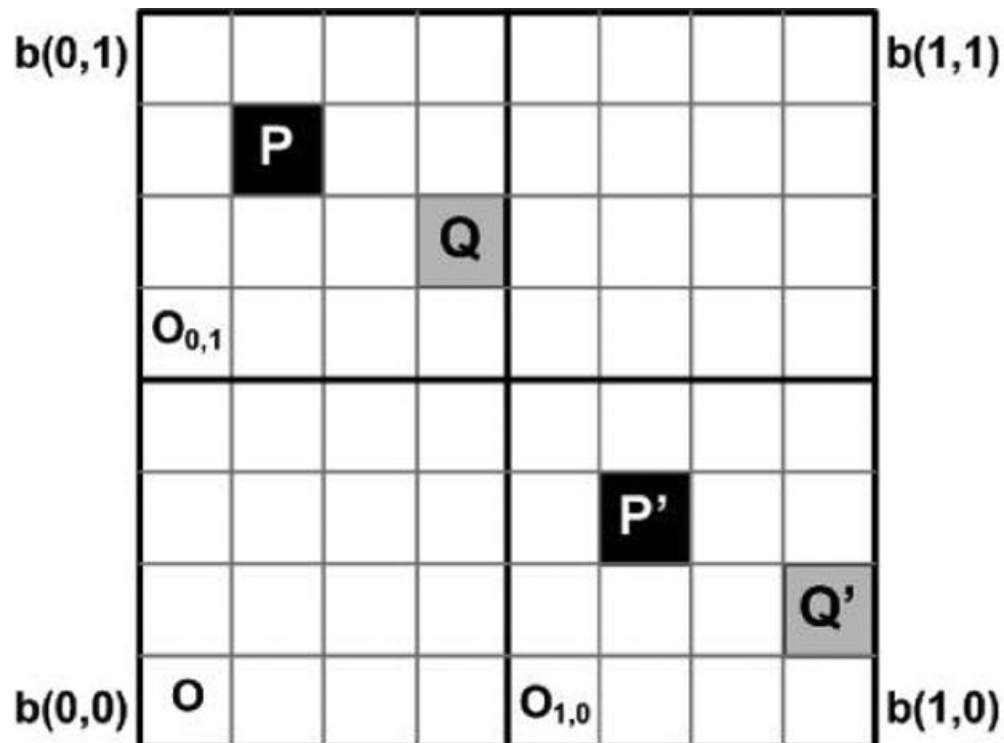
- Paralelizarea se face in modul urmator:
- Local: se considera punctele P si P' ce au aceeasi pozitie relativa la originea blocului din care fac parte \rightarrow TH de P relativ la originea blocului $O(0,1)$ este egala cu TH de P' relativ la originea blocului $O(1,0)$





Tehnica de paralelizare

- Global: se foloseste proprietatea de aditivitate pentru fiecare bloc, obtinandu-se imaginea finala.





Implementare hardware

- Prin paralelizare, respectand proprietatile enuntate, la sfarsitul unui ciclu se calculeaza $k*k$ blocuri:

$$\begin{aligned} HT \left(P \left(x + i \left(\frac{m}{k} \right), y + j \left(\frac{m}{k} \right) \right), O \right)_{i,j=0,1,\dots,k-1} \\ = HT(P(x, y), O_{b(i,j)}) + HT(O_{b(i,j)}, O) \end{aligned}$$



Implementare hardware

- Din punct de vedere computational timpul total al algoritmului clasic se exprima dupa formula:

$$T' = (1 + p\alpha) m^2$$

- Unde p este raportul dintre nr. de pixeli de margine si suprafata totala a imaginii, iar α este timpul aditional daca un pixel este o margine



Implementare hardware

- Pentru algoritmul paralelizat timpul se exprima dupa formula:

$$T_{prop} = \left(\frac{m}{k}\right)^2$$

- Unde k este dimensiunea unui bloc.

Comparativ

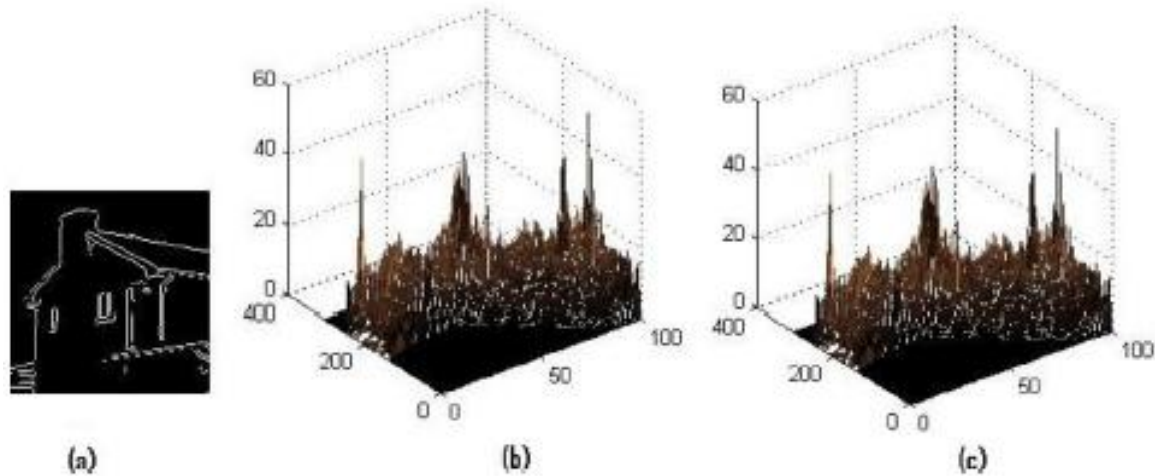


Fig. 5. (a) Edge map of benchmark image “house.png.” (b) Hough transform using conventional method. (c) Hough transform using AHT.

TABLE I
TOTAL COMPUTATION TIME FOR 128×128 IMAGE

Design	T cycles
Proposed AHT based	256
Iterative CORDIC based [3], [4]	1,240,008
Pipelined CORDIC based [3], [4]	105,668
Pipelined CORDIC based [5]	105,668
Repeated addition based [6]	17,234



Concluzii

- Pentru algoritmul propus, aplicat pe o imagine de 128x128 cu 5% continut, rezultatele au fost de 67 ori mai rapide.
- Pentru o imagine mai mare, timpul poate fi redus prin gasirea unui raport bun intre dimensiunea imaginii si dimensiunea unui bloc