



# Képfeldolgozás – jól párhuzamosítható

B. Wilkinson, M. Allen: „Parallel Programming”, Pearson Education Prentice Hall, 2nd ed., 2005. könyv 12. fejezete alapján



# Vázlat

- A képfeldolgozás olyan alkalmazási terület, amely számos lehetőséget biztosít párhuzamosításra

Témakörök:

- Alacsonyszintű műveletek: simítás, hisztogram, éldetektálás
- Hough transzformáció egyenes detektáláshoz



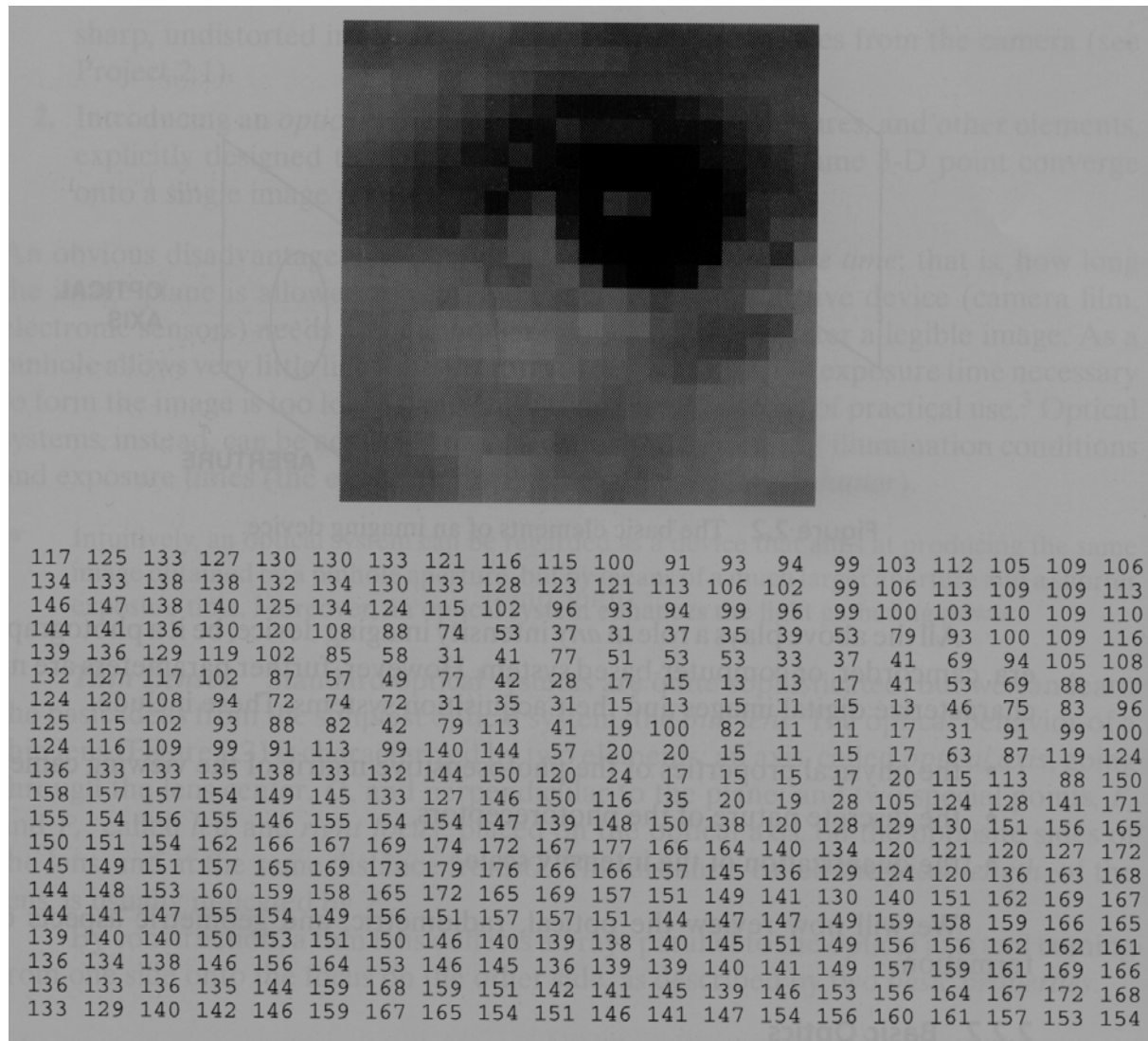
# Mi a képfeldolgozás?

- A képfeldolgozás a jelfeldolgozás része, amely képekkel foglalkozik
- Célja: a kép minőségének javítása az ember, vagy további számítógépes feldolgozás számára
- Kép – Képfeldolgozás – “Jobb” kép

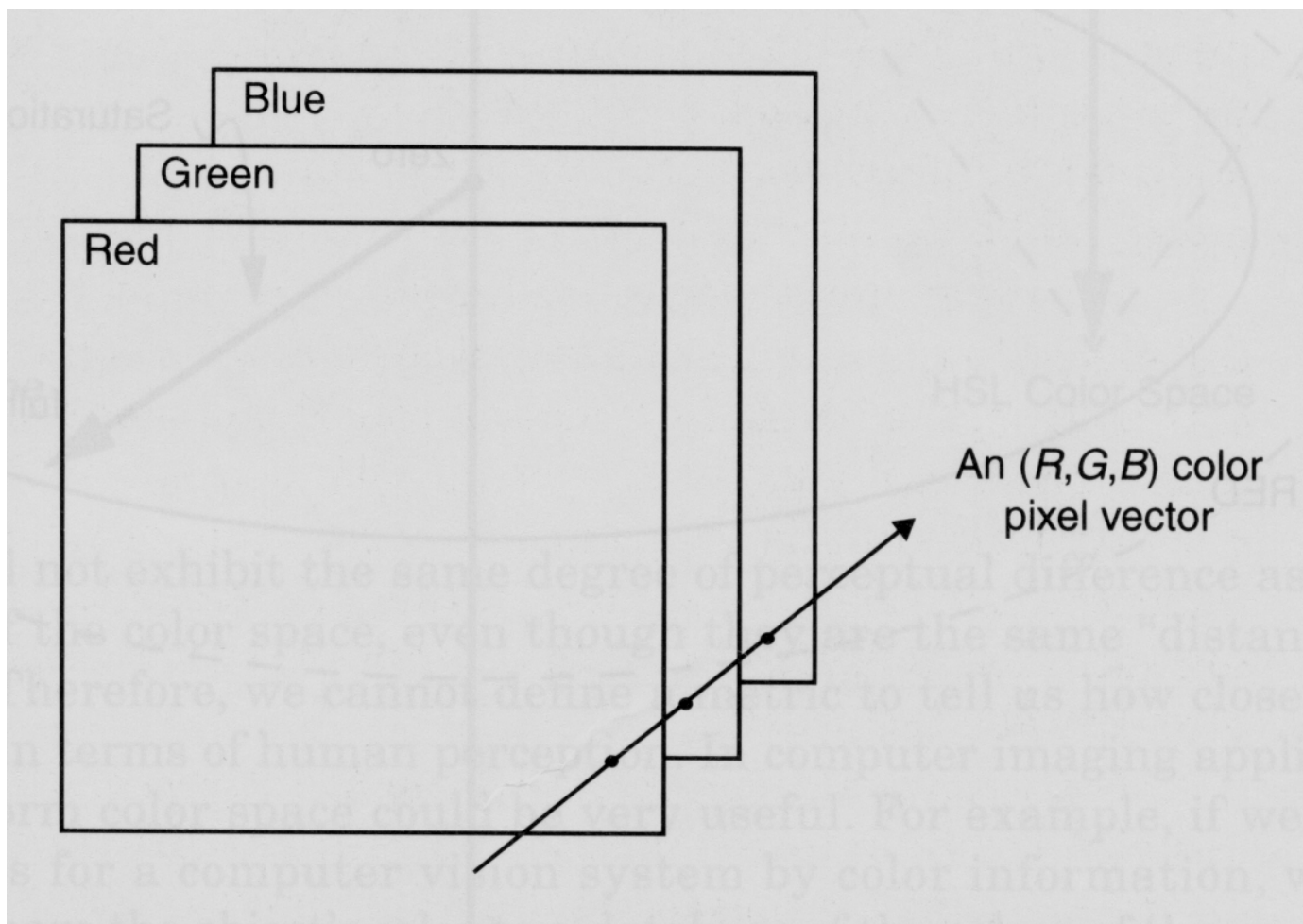
# Képfeldolgozás nehézségei



# Képek ábrázolása



# Színes képek



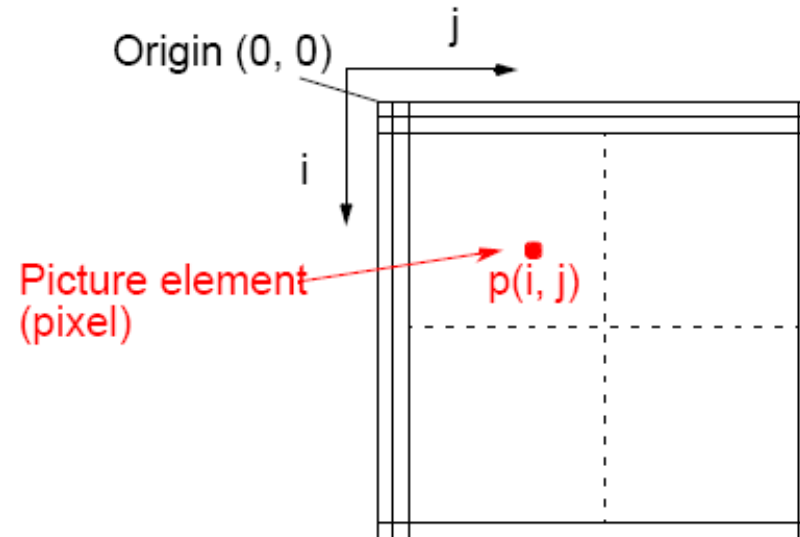


# A digitális képfeldolgozás szintjei

- A képek szg-es feldolgozását három szintre lehet osztani: alacsony, közép és magas szintű feladatok (low-level, middle-level, high-level)
- Low-level: mind az input mind az output kép
- Middle-level: az inputok általában képek, de az outputok a képekből nyert attribútumok (pl. egy objektum azonosítói a képen)
- High-level: a felismert objektumok együttesének érzékelése

# Alacsonyszintű képfeldolgozás

- A tárolt képen operál, hogy javítsa/módosítsa azt
- A kép képelemek (pixel = picture element) kétdimenziós tömbje



- Számos alacsonyszintű képművelet az intenzitásokat (szürkeségi értékeket) manipulálja





# Számítási igény

- Tételezzünk fel egy  $1024 \times 1024$  pixelből álló képet, ahol az intenzitást 8-biten tároljuk.
- Tárolási igény  $2^{20}$  byte (1 Mbytes)
- Tegyük fel, hogy minden pixelen csak egy műveletet végzünk, ekkor  $2^{20}$  operációt kell végeznünk a képen. Ez kb.  $10^{-8}$  mp/operáció, ami hozzávetőlegesen 10 ms.
- Valós idejű képfeldolgozás esetén tipikusan 25-30 képet kell másodpercenként feldolgozni (fps).
- Tipikusan nem csak egyetlen operációt kell végezni a pixeleken, hanem több és összetettebb funkciókat.



# Pont alapú műveletek

Olyan műveletek, ahol az output csak egy képponttól függ.

*Küszöbölés:*

- Egy bizonyos határnál (threshold) nagyobb intenzitású képpont értékéhez az egyik szélső értéket, ellenkező esetben a másik szélsőértéket rendeljük hozzá.

Ha egy pixel intenzitása  $x_i$ , akkor minden pixelre:

if ( $x_i < \text{threshold}$ )  $x_i = 0$ ; else  $x_i = 1$ ;



# Pont alapú műveletek

*Kontraszt nyújtás:*

- Az intenzitás értékek tartományát széthúzzuk, hogy a részletek jobban érzékelhetőek legyenek. Adott egy pixel intenzitása  $x_i$  az  $x_l - x_h$  tartományból, a kontraszt nyújtás során az  $x_H - x_L$  tartományba transzformáljuk az intenzitást

$$x_I = \frac{(x_H - x_L)}{(x_h - x_l)} x_i + x_L$$

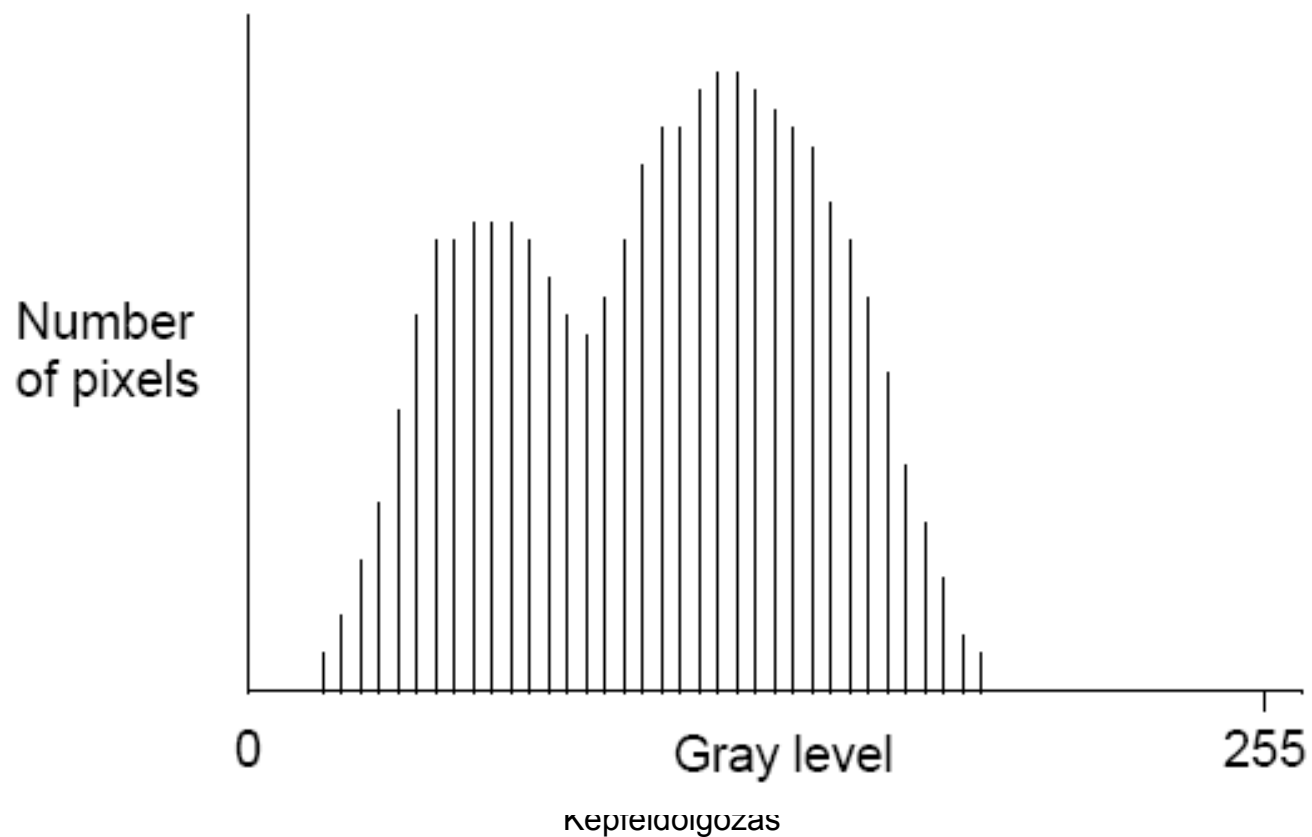
*Szűrkeségi szint csökkentés:*

- A kevésbé szignifikáns biteket elhagyjuk.



# Hisztogram

- Az egyes intenzitásokból hány darab van a képen





# Hisztogram soros kód

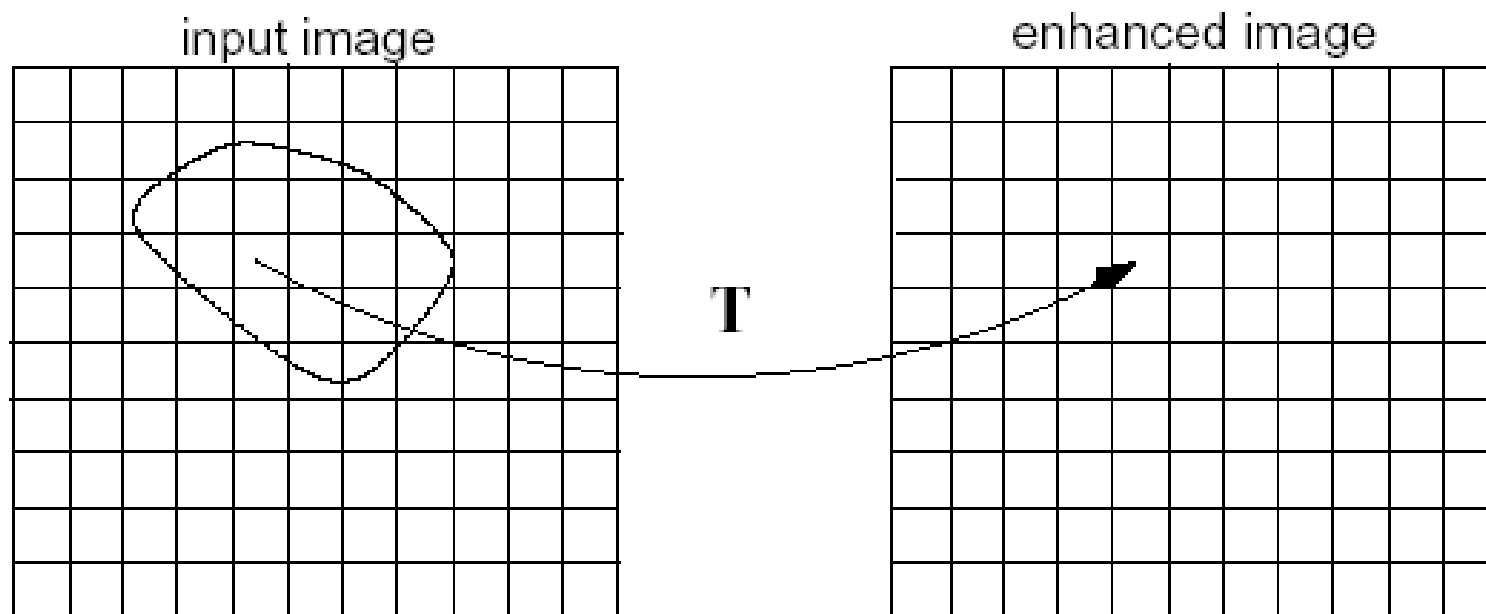
```
for(i = 0; i < height_max; x++)  
    for(j = 0; j < width_max; y++)  
        hist[p[i][j]] = hist[p[i][j]] + 1;
```

ahol a pixeleket a `p[i][j]` tömb tárolja és a `hist[k]` vektor megmondja, hogy a  $k$ -ik intenzitásból hány darab van a képen.

- Egyszerű összegző tömb.
- Könnyen párhuzamosítható dekompozícióval.

# Maszk alapú műveletek

- $g(x, y) = T[f(x, y)]$ ,  $T$  a szomszédos pixeleken operál (lokális művelet)





# Maszk alapú műveletek

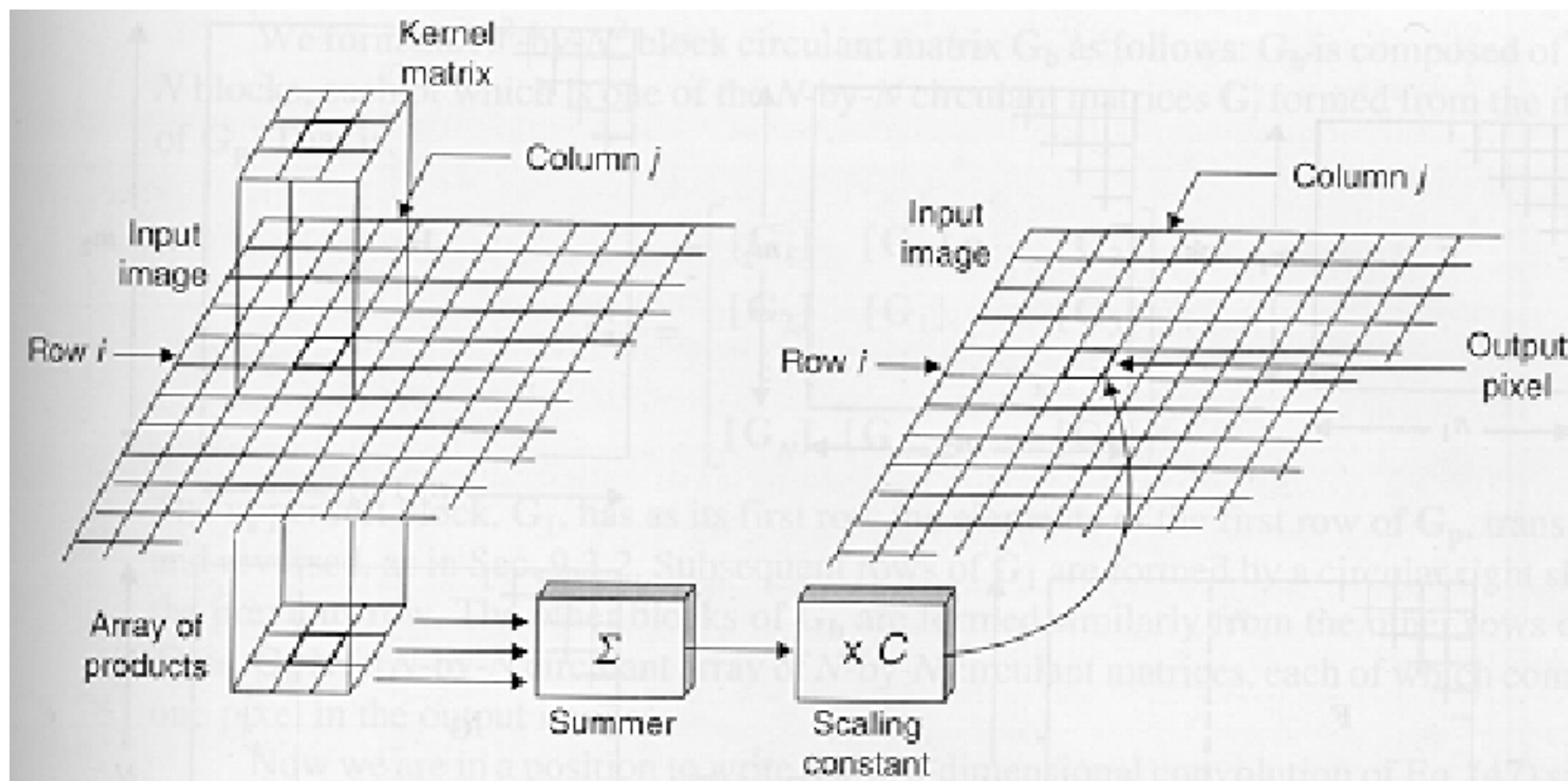
*Simítás (zajszűrés):*

- Az intenzitás nagy változásait simítjuk el, a magas-frekvenciás tartalom csökkentése

*Élesítés:*

- A részletek kiemelése

# Maszk használata







# Maszk

- Ablak alapú műveletekhez gyakran alkalmaznak  $n \times n$  méretű maszkot ( $n = 3, 5, 7, \dots$ )

$x_0$	$x_1$	$x_2$
$x_3$	$x_4$	$x_5$
$x_6$	$x_7$	$x_8$



# Átlagoló szűrő

- Egyszerű simítási technika, ahol az ablakban lévő intenzitások átlaga az új intenzitás érték:

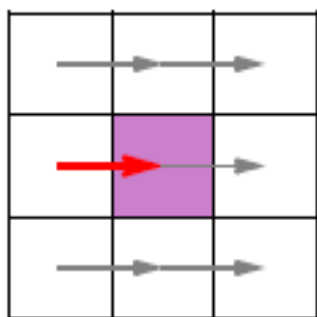
$$x'_4 = \frac{x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8}{9}$$

- Soros kód:  
Kilenc (+1) lépés kell az átlag kiszámításához,  $n$  pixelre  $9n$ . Komplexitás:  $O(n)$ .

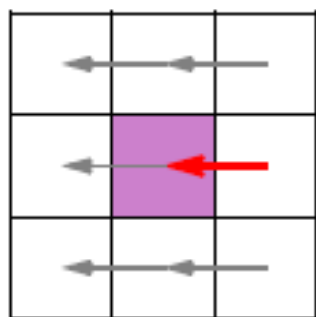


# Párhuzamos átlagoló szűrő

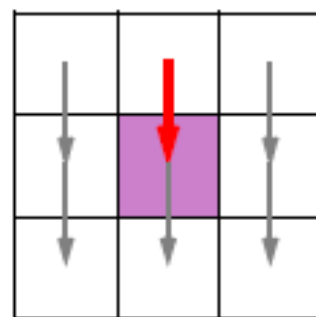
- A műveletek száma redukálható négyre
  - Elv, ami pontosításra kerül mindjárt



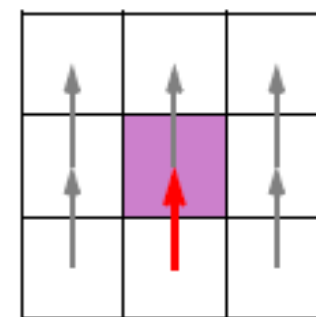
Step 1  
Each pixel  
adds pixel  
from left



Step 2  
Each pixel  
adds pixel from  
right

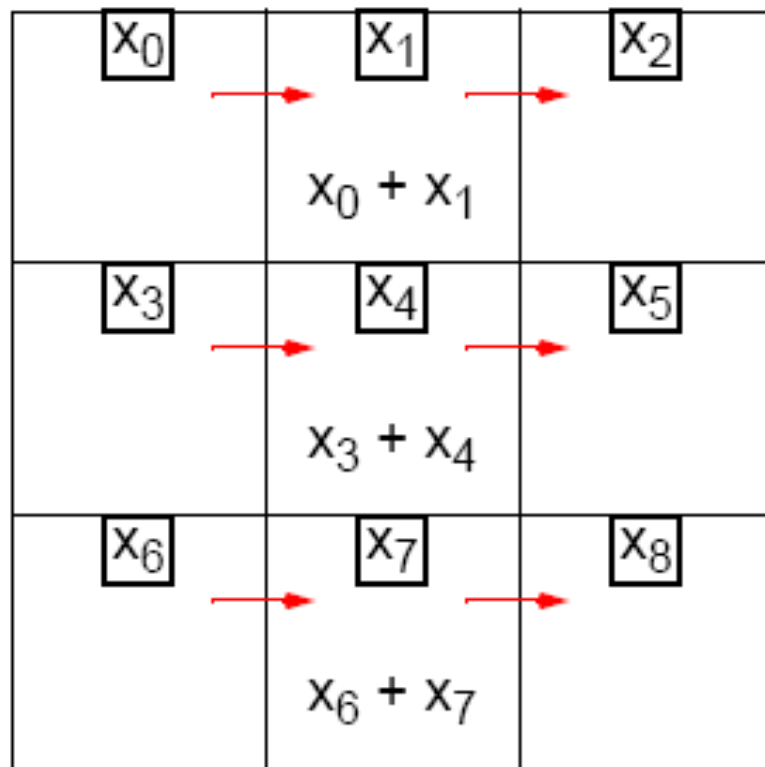


Step 3  
Each pixel adds  
pixel from above

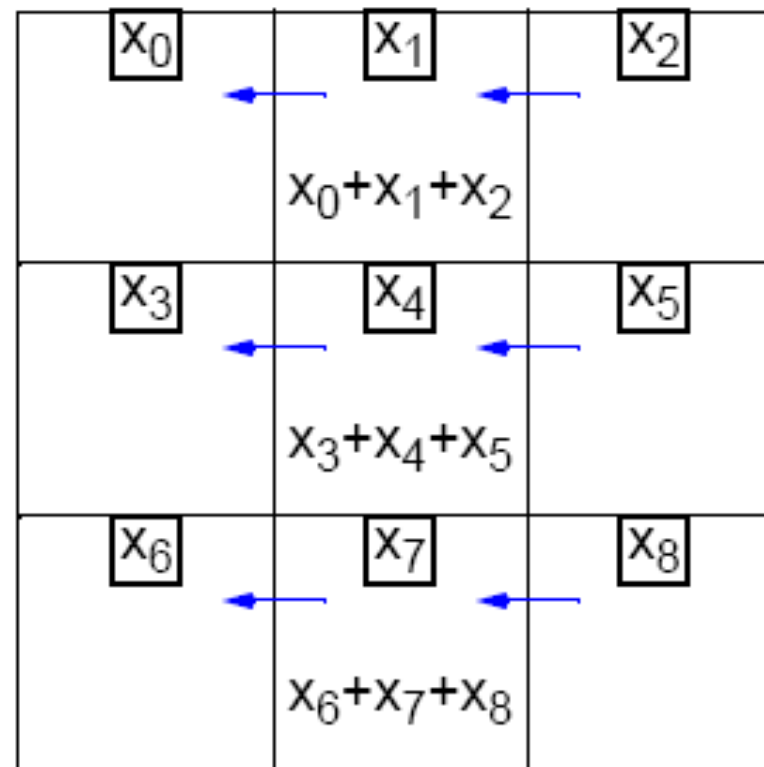


Step 4  
Each pixel adds  
pixel from below

# Párhuzamos átlagoló szűrő I.

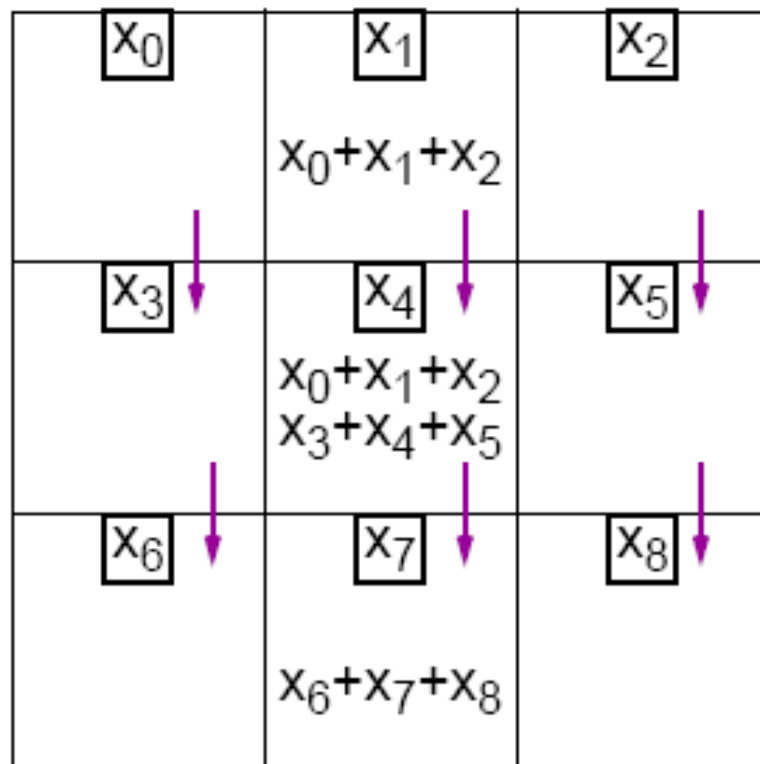


(a) Step 1

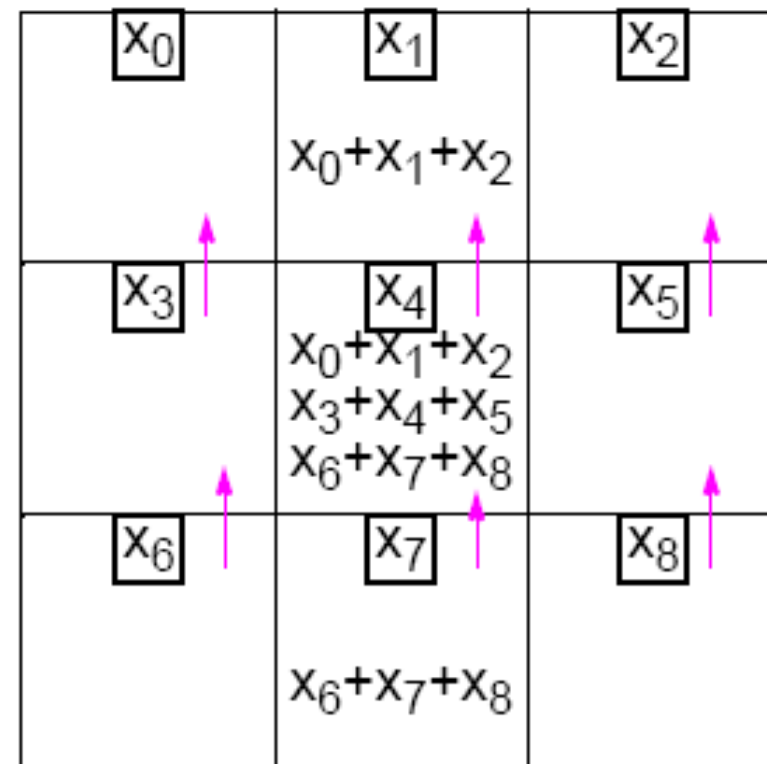


(b) Step 2

# Párhuzamos átlagoló szűrő II.



(c) Step 3



(d) Step 4



# Medián szűrő

Soros megvalósítás:

- A medián meghatározása érdekében rendezni kell a pixelértékeket és a középsőt kell kiválasztani.
  - Például 3 x 3-as esetben  $y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7$ , és  $y_8$ .  
A medián  $y_4$ .
  - Az ötödik elemet kell kivenni a rendezés után.
  - Pl. buborékos rendezésnél a műveletek (összehasonlítás és ha kell csere) száma:  $8 + 7 + 6 + 5 + 4 = 30$  lépés, azaz  $n$  pixelre  $30n$  művelet.

# Közelítő medián szűrő

Párhuzamos megvalósítás:

- Elsőként a soron belül hajtsunk végre három összehasonlítást és cserét:

$$p_{i,j-1} \leftrightarrow p_{i,j}$$

$$p_{i,j} \leftrightarrow p_{i,j+1}$$

$$p_{i,j-1} \leftrightarrow p_{i,j}$$

ahol  $\leftrightarrow$  jelenti, hogy hasonlítsd össze és cseréld fel, ha a baloldali érték nagyobb, mint a jobboldali.

- Ezután oszlopokra vonatkozóan három lépés:

$$p_{i-1,j} \leftrightarrow p_{i,j}$$

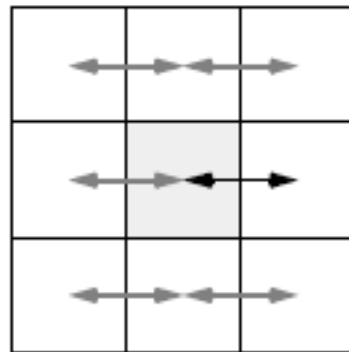
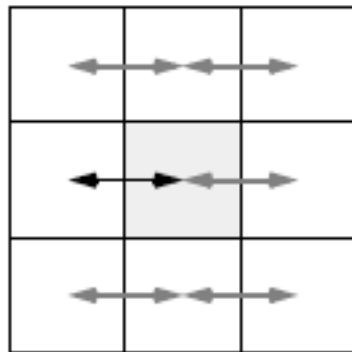
$$p_{i,j} \leftrightarrow p_{i+1,j}$$

$$p_{i-1,j} \leftrightarrow p_{i,j}$$

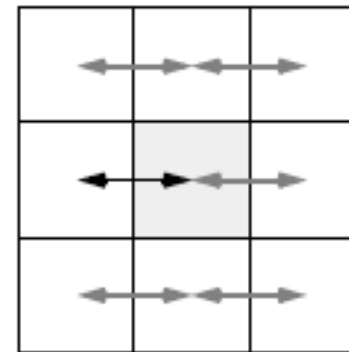
- Összesen hat lépés
- Mikor nem pontos?



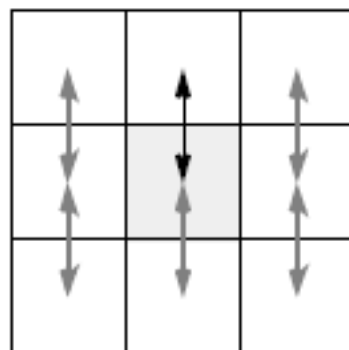
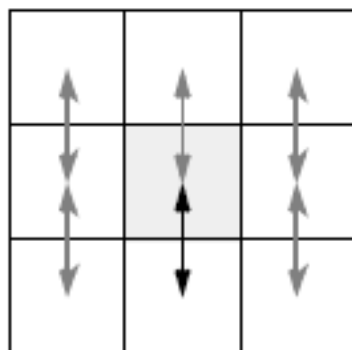
# Közelítő medián szűrő



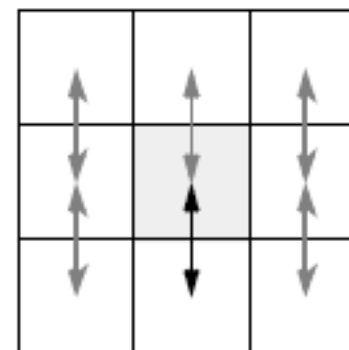
Largest  
in row



Next largest  
in row



Next largest  
in column







# Súlyozó maszkok

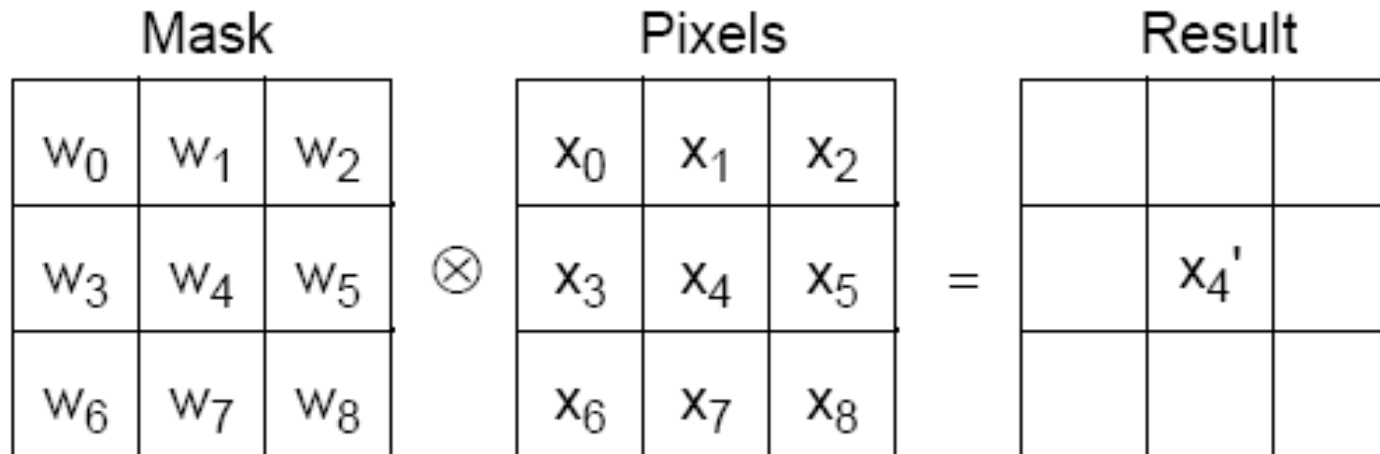
- Gyakran nem egységnyi súlyúak a maszkelemek:  $w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7$ , és  $w_8$ .  
Az új intenzitás érték  $x_4'$ :

$$x_4' = \frac{w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6 + w_7 x_7 + w_8 x_8}{k}$$

ahol  $1/k$  skálázási tényező az intenzitás értékét állítja be és  $k$  nagysága gyakran a súlyok összegével egyezik meg.

# Kereszt-korreláció

- 3 x 3-as súlyozó maszk használatával



azaz a súlyozott összeg ( $w_i x_i$ ) két függvénynek ( $f$  és  $w$ ) a (diszkrét) kereszt-korrelációja:  $f \otimes w$

# Maszkok

- Átlagoló

$k = 9$

1	1	1
1	1	1
1	1	1

- Zajszűrő

$k = 16$

1	1	1
1	8	1
1	1	1

- Felüláteresztő élesítő

$k = 9$

-1	-1	-1
-1	8	-1
-1	-1	-1

# Éldetektálás

- Ahol az intenzitásban jelentős változás van, ott található él.

- Kép deriváltja:

$$\frac{\partial f(x, y)}{\partial x} = \lim_{\varepsilon \rightarrow 0} \left( \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \right)$$

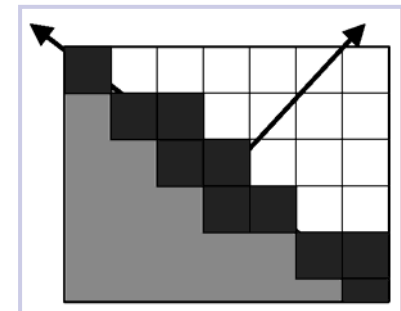
$$\frac{\partial f(x, y)}{\partial y} = \lim_{\varepsilon \rightarrow 0} \left( \frac{f(x, y + \varepsilon) - f(x, y)}{\varepsilon} \right)$$

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{f(x_{n+1}, y_m) - f(x_n, y_m)}{\Delta x}$$

$$\frac{\partial f(x, y)}{\partial y} \approx \frac{f(x_n, y_{m+1}) - f(x_n, y_m)}{\Delta y}$$

- Gradiens:  $\text{grad}(f) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

Képfeldolgozás



# Éldetektálás differenciálással

- Intenzitás



- Első derivált



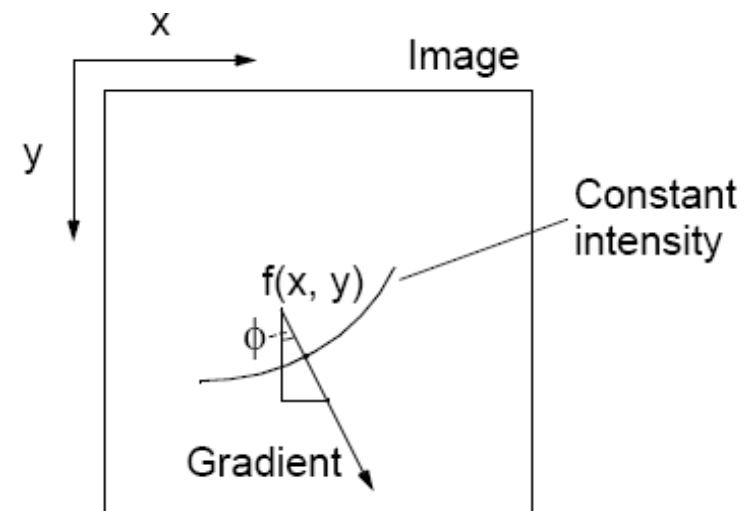
- Második derivált



# Gradiens nagyság és irány

- Gradiens nagyság:  $|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

- Gradiens irány:  $\varphi(x, y) = \arctan \left\{ \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right\}$



- Gradiens nagyságra más mérték:  $|\nabla f| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$

# Éldetektáló maszkok

- A gradiens összetevőinek számolása:

$$\left| \frac{\partial f}{\partial x} \right| = (x_5 - x_3) \quad \left| \frac{\partial f}{\partial y} \right| = (x_7 - x_1)$$

- A gradiens nagyság:  $|\nabla f| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| = |x_7 - x_1| + |x_5 - x_3|$

0	-1	0
0	0	0
0	1	0

0	0	0
-1	0	1
0	0	0

# Prewitt operátor

- A gradiens összetevőinek számolása:

$$\left| \frac{\partial f}{\partial x} \right| = (x_2 - x_0) + (x_5 - x_3) + (x_8 - x_6)$$

$$\left| \frac{\partial f}{\partial y} \right| = (x_6 - x_0) + (x_7 - x_1) + (x_8 - x_2)$$

- A Prewitt maszk:

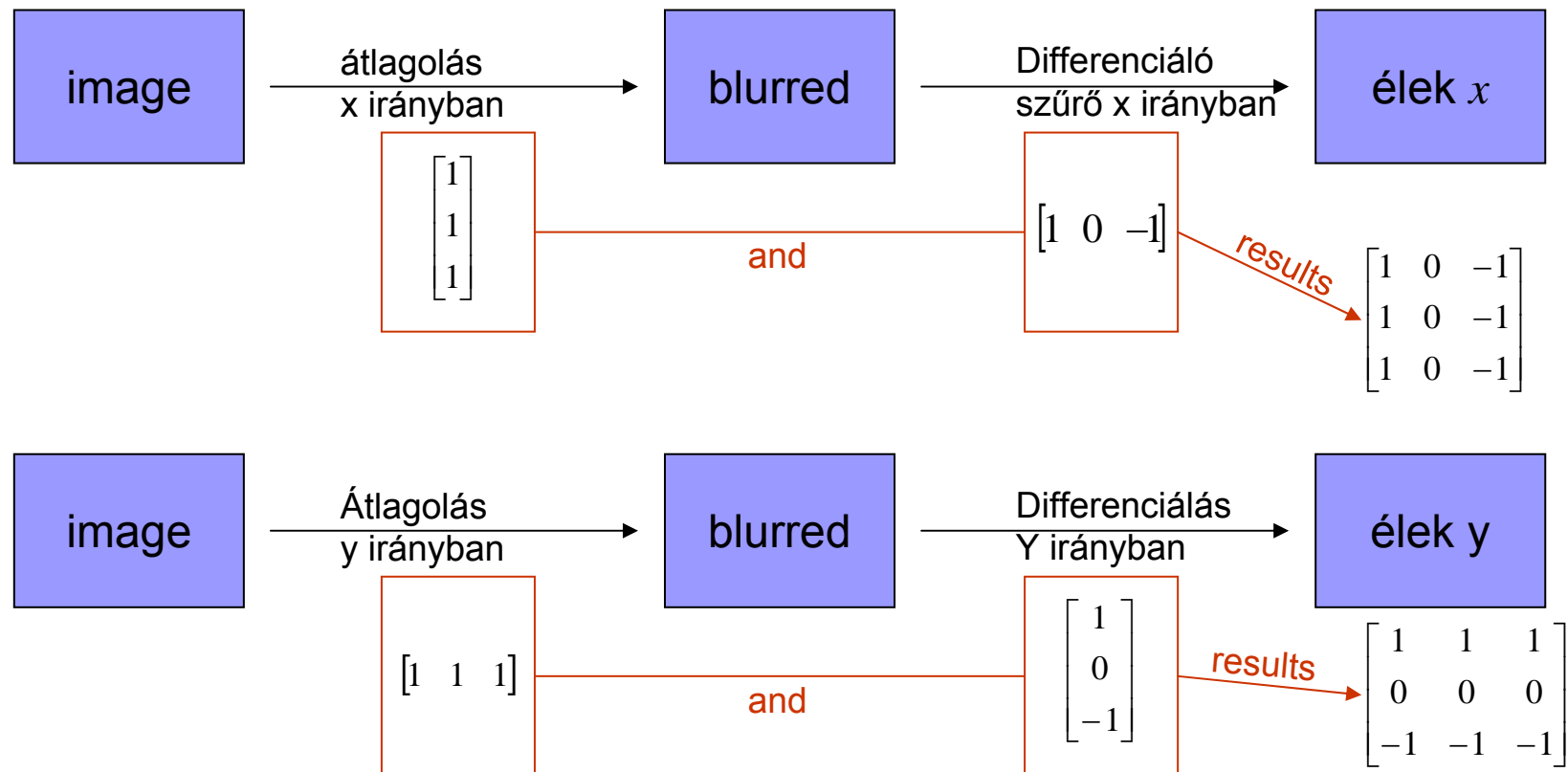
-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

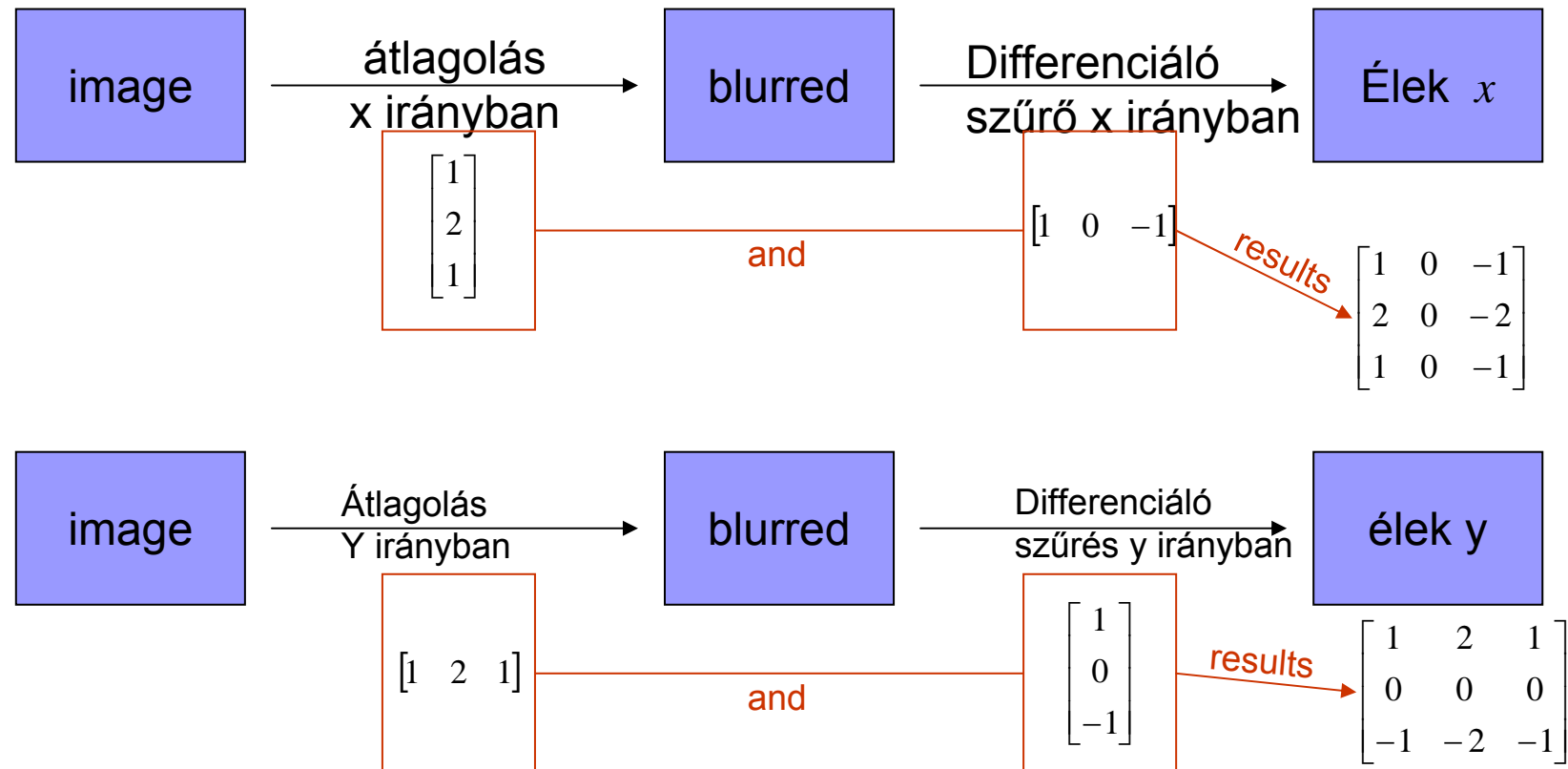
reprezentáció



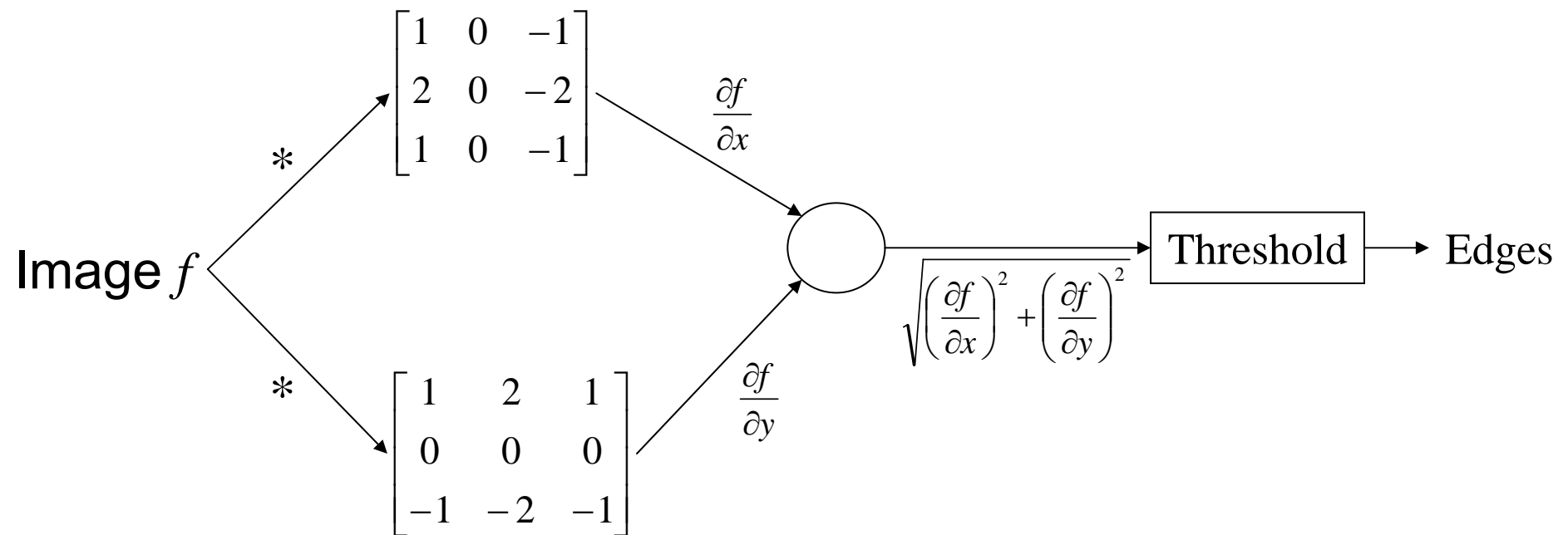
# Prewitt éldetektor



# Sobel éldetektáló



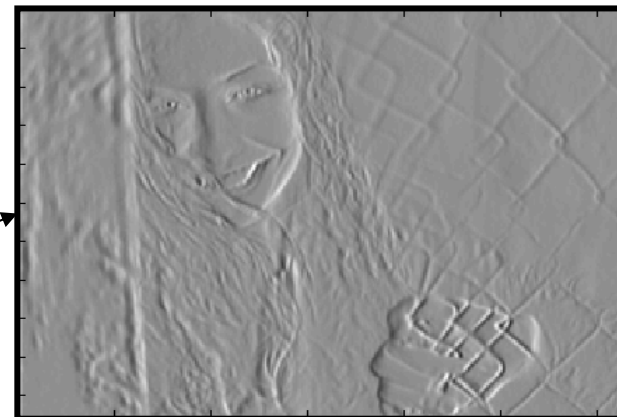
# Sobel éldetektáló



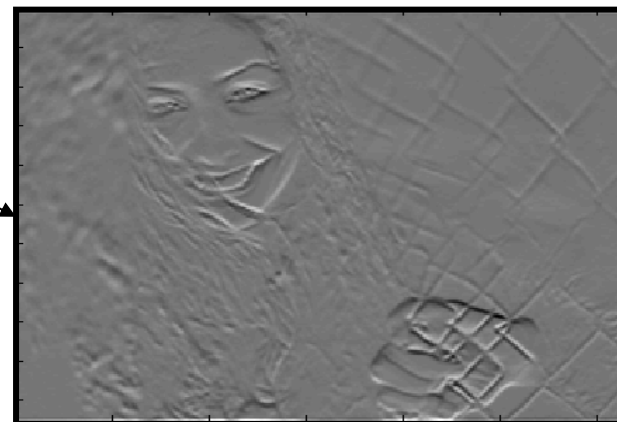
# Sobel éldetektáló



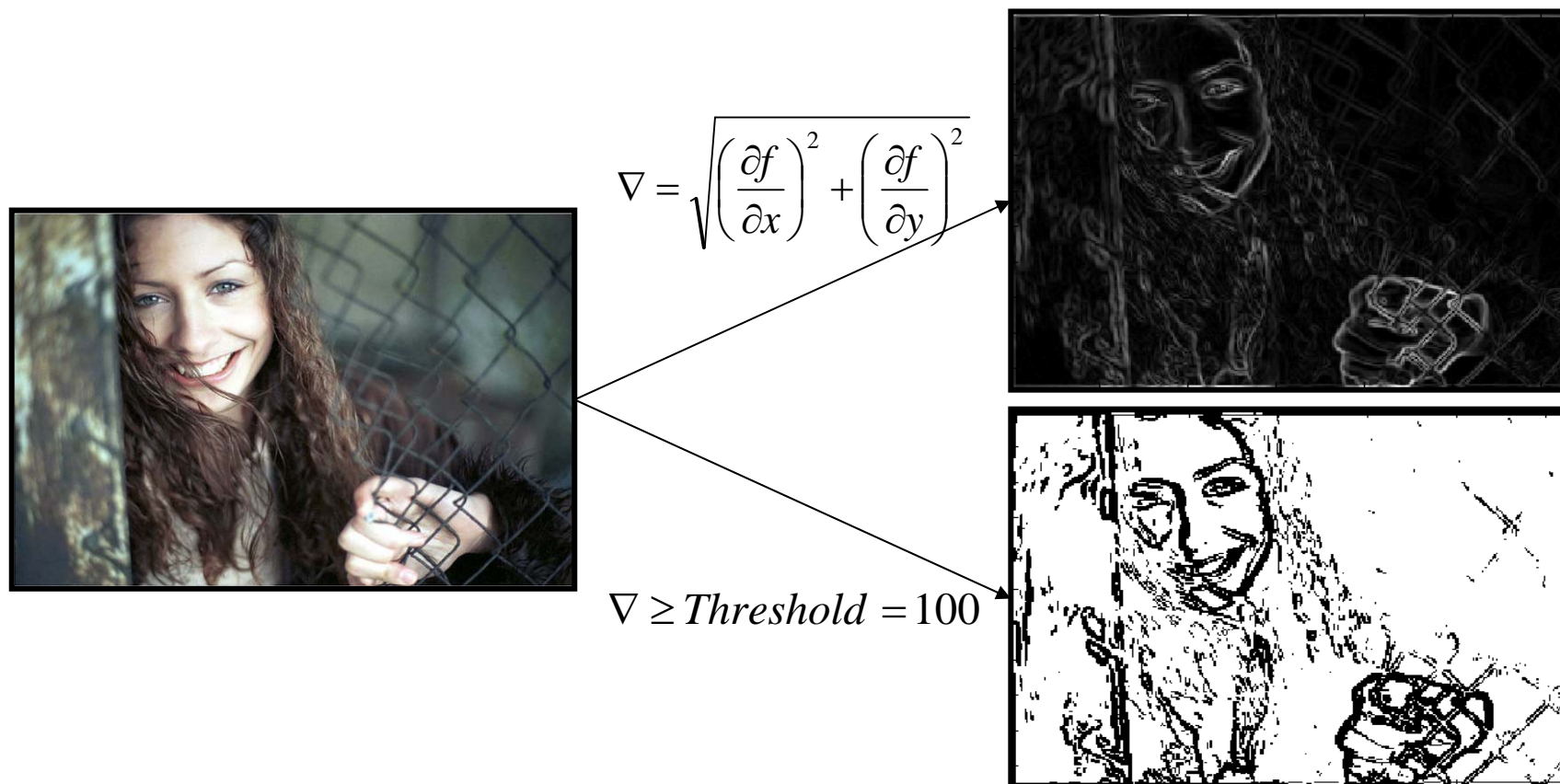
$$\frac{\partial f}{\partial x}$$



$$\frac{\partial f}{\partial y}$$

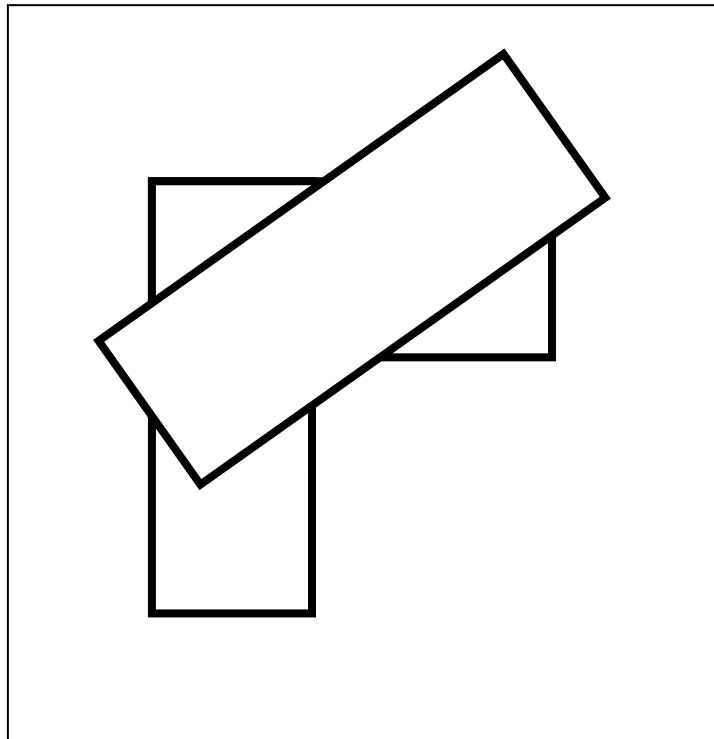


# Sobel éldetektáló



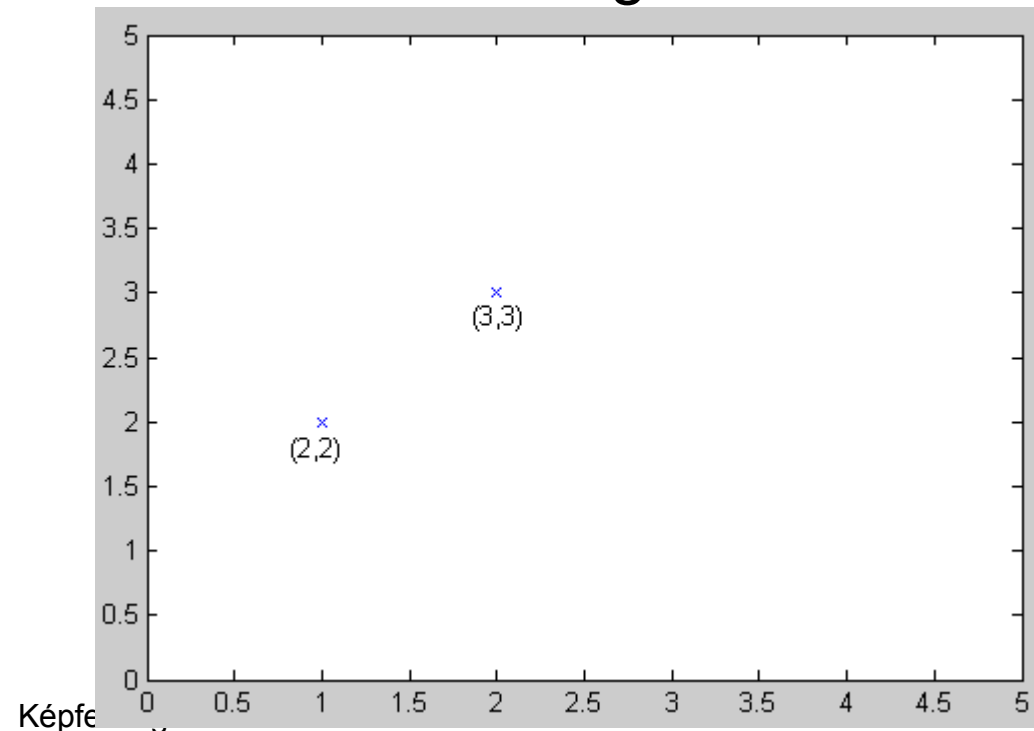


# Vonaldetektálás



# Vonaldetektálás Hough trafóval

- Cél: Az  $f(x, y)$  képen találjuk meg a vonalakat és határozzuk meg azok egyenletét
- $O(NNMM)$  nagyságrendű számítást kell elvégezni





# Vonaldetektálás Hough trafóval

Főbb pontok:

- Konverzió paraméter térbe
- Az  $m$  és  $n$  paraméterek megtalálása
- Visszakonvertálás derékszögű koordinátákba



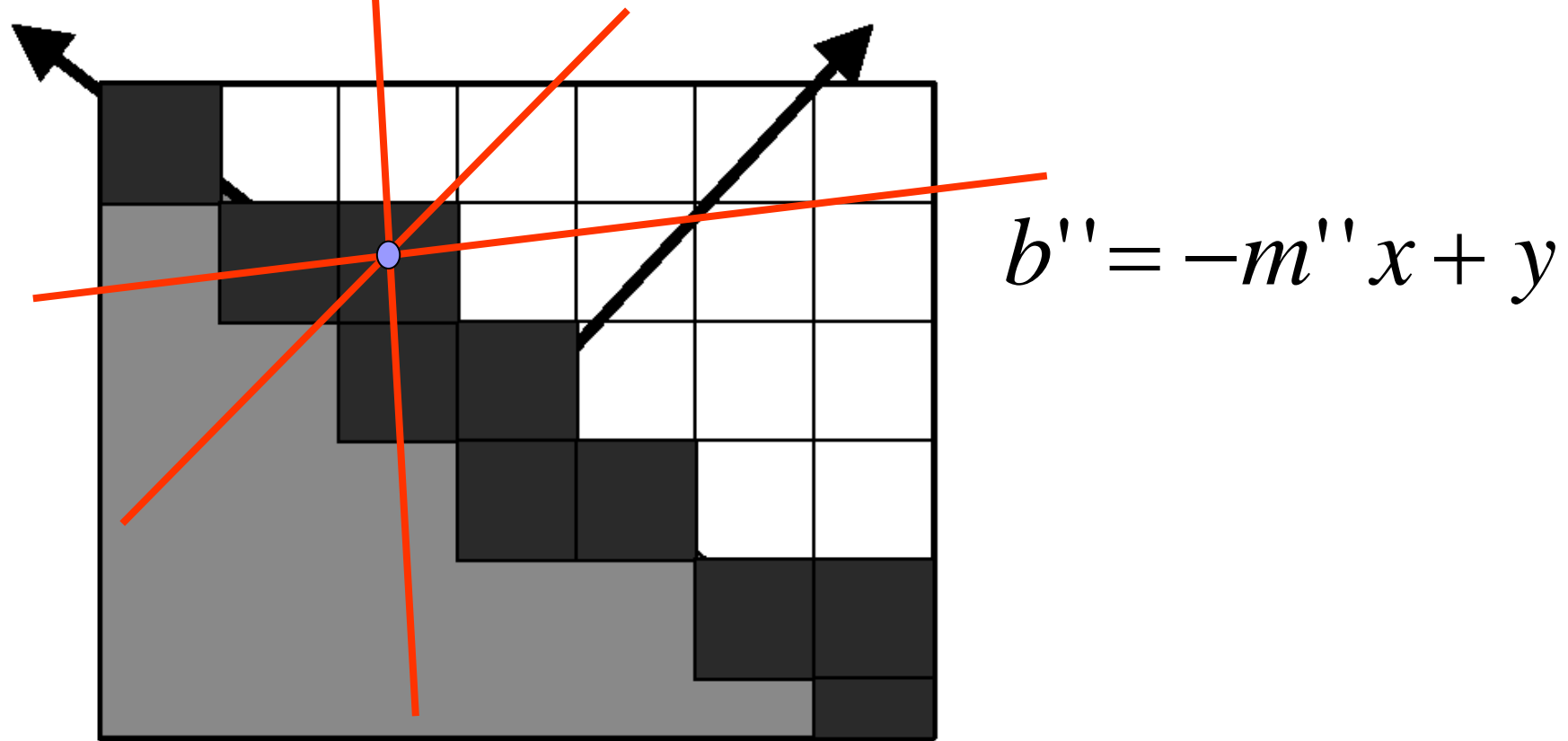


# Vonaldetektálás Hough trafóval

- **Kulcs:** Használjuk a paraméterteret, ahol a bonyolult probléma az egyszerűbb lokális maximumok megtalálását jelenti
- Input:
  - ☐ Bináris kép élpontokkal
  - ☐ Küszöb

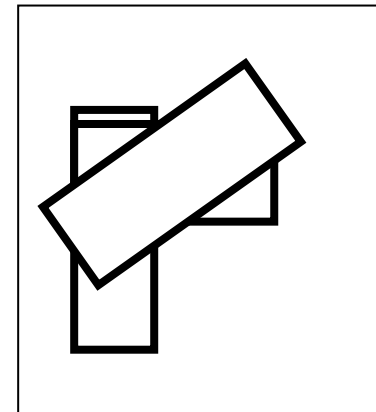
# Hough transzformáció

$$b = -mx + y \quad b' = -m'x + y$$



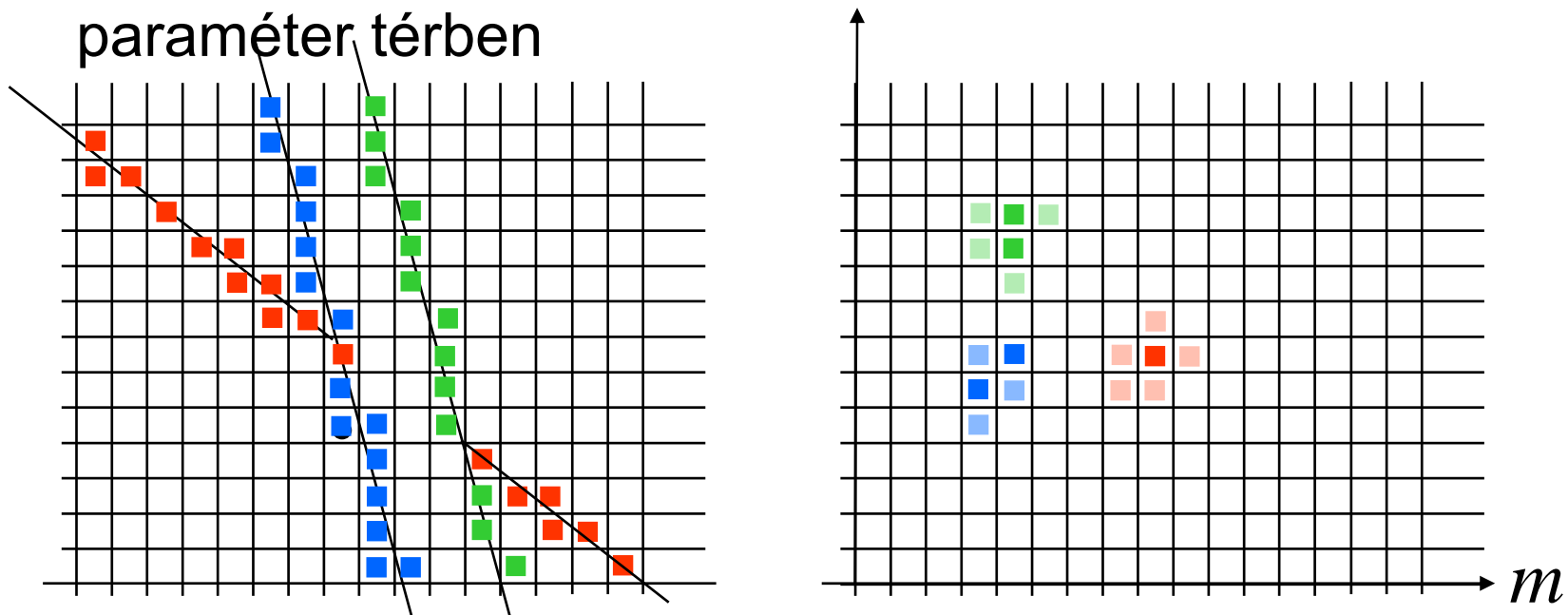
# Vonalillesztés

- Vonal egyenlet  
 $y = mx + b$   $m$  meredekség,  $b$  az  $y$  - metszéspon t
- Az  $(m, b)$  teret osszuk fel egy ráccsal és minden cellához rendeljünk egy számlálót:  $c(m, b)$   
kezdetben 0 értékkel
- Minden élpixel ismert koordinátaival
  - Számoljuk ki  $b$  értékét minden lehetséges  $m$  mellett  
 $b_i = y - m_j x$
  - Növeljük meg a  $c(m_i, b_i)$ -t eggyel
- Keressük meg a lokális maximumokat a paraméter térben!



# Hough trafó: kvantálás

- Vonal detektálás maximum/klaszter keresésével a paraméter térben



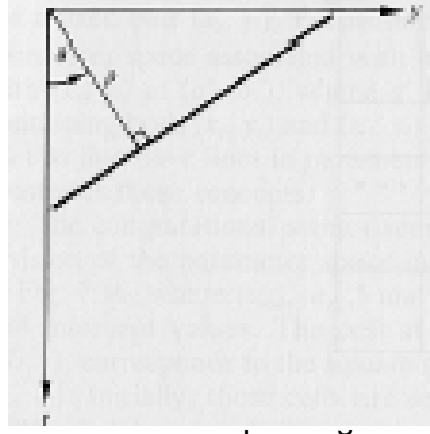
- Függőleges vonalak esetén probléma
  - $m$  és  $b$  végtelen

# Hough transzformáció

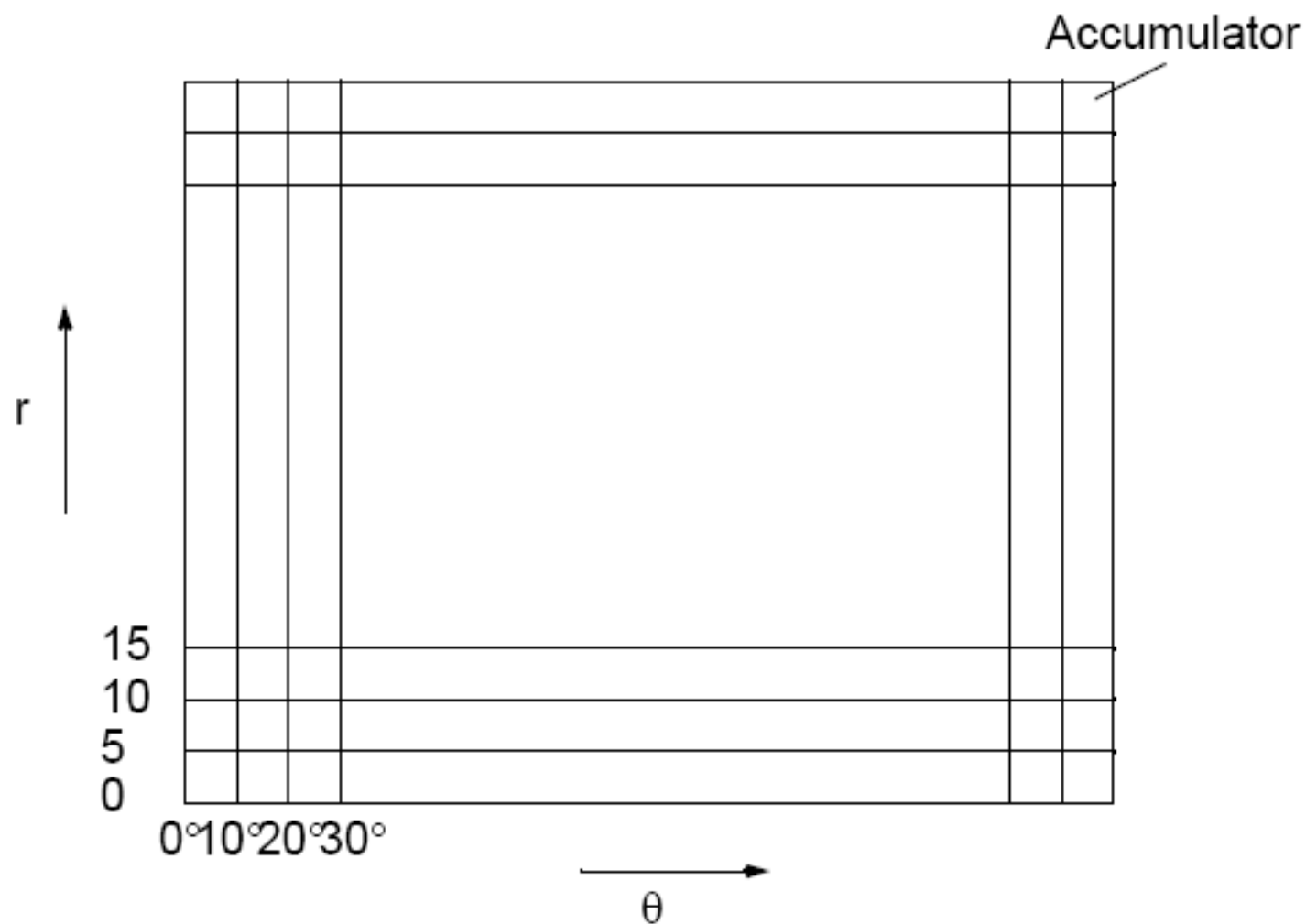
- Polár koordinátás reprezentáció
  - Egy egyenes minden pontjára  $\theta$  és  $\rho$  állandó
  - Bármely irányban numerikusan stabil leírás

$$x \cos \theta + y \sin \theta = \rho$$

- Különböző  $\theta$  konstans értékekre  $\rho$  fix értékeinél különböző vonalakat szolgáltatnak



# Akkumulátor tömb

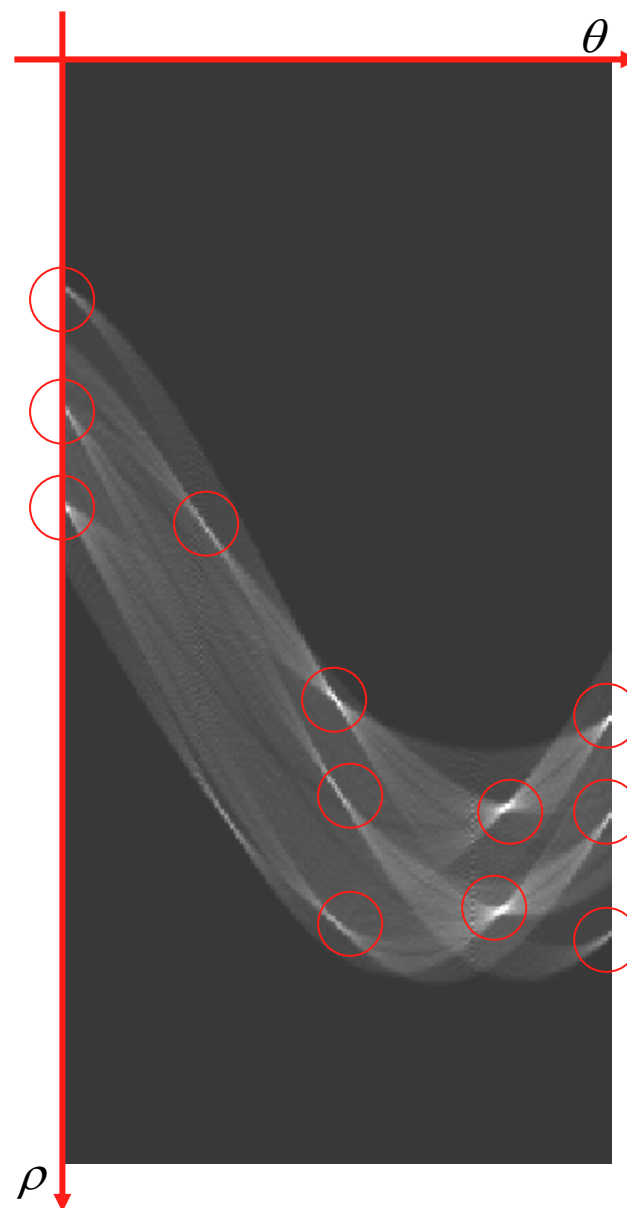
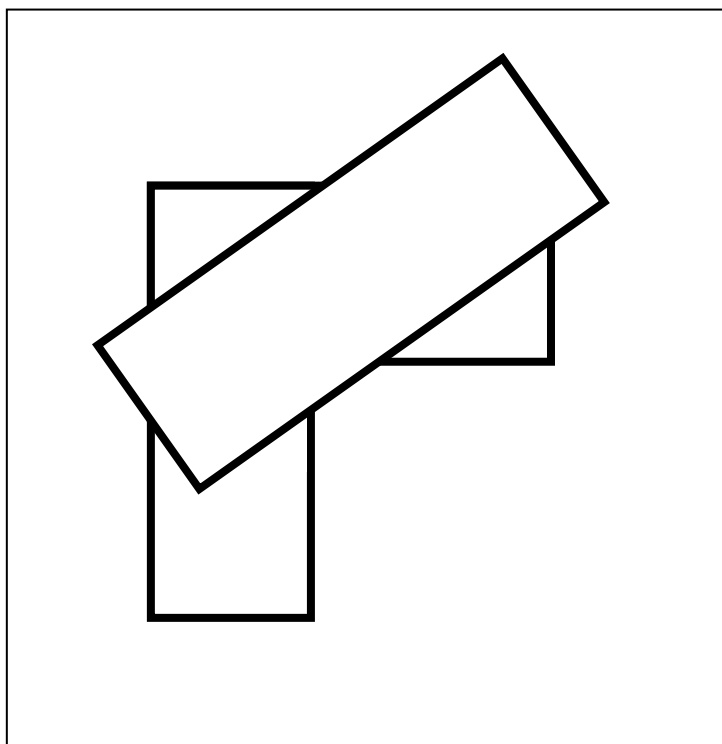




# Algoritmus

- Készítsünk egy 2D  $(\theta, \rho)$  számláló tömböt, a szög 0 és 180 fok között változik, a távolság maximum a kép átlója
  - Nullázuk ki
- A  $\theta$  szög lehetséges értékeit vegyük fel
  - Például 10°-os növekmények
- Minden élpontra
  - Számoljuk ki  $\rho$  értékét az (A) egyenlettel
  - Minden kiszámolt  $(\theta, \rho)$  párra növeljük meg a számlálótömb értékét
- Keressük meg a lokális maximumokat

# Vonaldetektálás

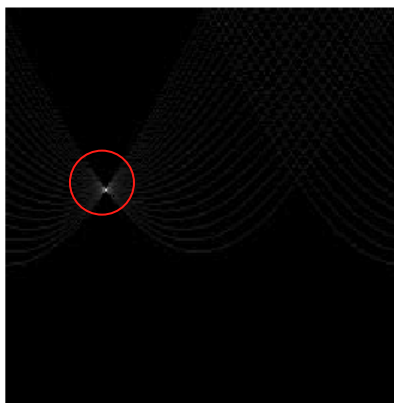
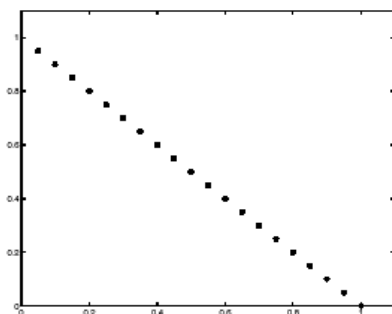


Képfeldolgozás  $\rho$

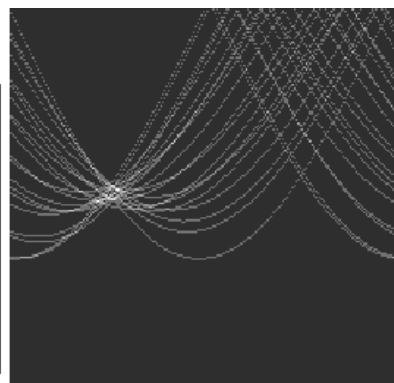
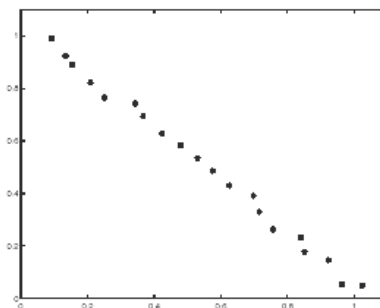


# Vonaldetektálás - példa

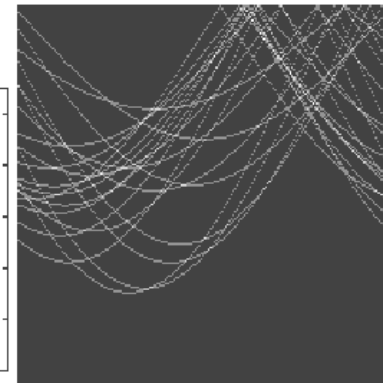
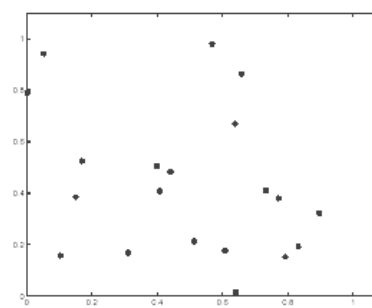
ideális



zajos



Nagyon zajos





# Nehézségek

- Hogyan osszuk fel a paraméter teret  $(\theta, \rho)$ ?
  - nagy? Nem tudunk különbséget tenni vonalak között
  - kicsi? A zaj hibákat eredményez
- Hány vonalat találunk?
- Melyik élpont melyik vonalhoz tartozik?
- A zaj miatt nehéz kielégítő megoldást találni



# Soros kód

```
for (x = 0; x < xmax; x++)                                /* for each pixel */
    for (y = 0; y < ymax; y++) {
        sobel(&x, &y, dx, dy);                            /* find x and y gradients */
        magnitude = grad_mag(dx, dy);                      /* find magnitude if needed */
        if (magnitude > threshold) {
            theta = grad_dir(dx, dy);                      /* atan2() fn */
            theta = theta_quantize(theta);
            r = x * cos(theta) + y * sin(theta);
            r = r_quantize(r);
            acc[r][theta]++;                                /* increment accumulator */
            append(r, theta, x, y);                        /* append point to line */
        }
    }
```



# Párhuzamosítás

- Mivel az akkumulátor tömb számítása független más összegzésektől, ezért párhuzamosítható:
  - Az egész képre olvasási jog szükséges.